

PROGRAMMEZ !

Le magazine du développement

# PROgrammez !

www.programmez.com

mensuel n°123 - octobre 2009

# Windows 7

## Guide du développeur



- ✓ Découvrez le pack API, les nouvelles fonctions
- ✓ Compatibilité et migration du code
- ✓ Multitouch et biométrie : simple comme bonjour !

### RECRUTEMENTS ET FORMATIONS

DÉVELOPPEURS, surfez sur la

## CRISE !



Nicolas Sadirac, Epitech

### BASE de DONNÉES

Les moteurs de stockage dans

## MySQL

## HTML 5

La révolution ?

## PHP et FLASH

Intégrer une webcam côté serveur



## WEBMASTER

Tester ses pages web

**Embarqué**  
Micro Framework :  
.Net sans Windows !

**Linux**  
Mettre en oeuvre ptrace

**Web**  
Développer des extensions Drupal  
Spring : maîtriser les mécanismes de sécurité

**Recherche**  
Le modèle fractal

M 04319 - 123 - F : 5,95 €



# DÉVELOPPEZ 10 FOIS PLUS VITE

PLATEFORME PROFESSIONNELLE  
DE DÉVELOPPEMENT (AGL)

Windows, .Net, Java  
Windows 98, 2000, NT, 2003, XP, Vista, 2008

# WINDEV

DÉVELOPPEZ 10 FOIS PLUS VITE  
AVEC L'OUTIL DE DÉVELOPPEMENT  
ÉLU **LE PLUS PRODUCTIF**\*



**WINDEV 14** est l'environnement de développement totalement intégré (IDE, ALM), intégralement en français, réputé pour sa **puissance** et sa **facilité** d'utilisation.

**WINDEV 14 est livré complet:** 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, **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,

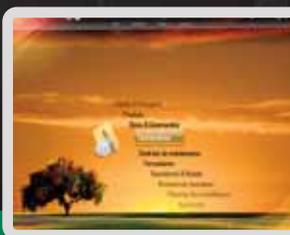
L5G, SNMP, Bluetooth, TAPI, OPC, FTP, HTTP, Socket, Twain, API, DLL, domotique, liaisons série et USB, débogage à distance, profiler, refactoring, **génère d'applications JAVA**, multilangue automatique, Gestionnaire de versions, Installateur 1-clic, etc...

Les applications créées fonctionnent avec toutes les versions de Windows: 98, 2000, NT, 2003, XP, Vista, sous TSE et Citrix, sur eeePC...

WINDEV 14 gère le **cycle complet de développement**, pour des équipes de 1 à 100 développeurs. Le support technique est **gratuit**\*

**Vous aussi, développez 10 fois plus vite... avec WINDEV 14.**

VERSION  
EXPRESS  
GRATUITE  
Téléchargez-la !



**VOTRE CODE EST  
MULTI-PLATEFORMES:**

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

Logiciel professionnel

**DEMANDEZ LE DOSSIER GRATUIT**

252 pages + DVD + Version Express incluse  
+ 112 Témoignages.

Tél: **04.67.032.032** ou **01.48.01.48.88**  
info@pcsoft.fr



Fournisseur Officiel de la Préparation Olympique

**www.pcsoft.fr**



Donnez votre avis sur ce numéro : [www.programmez.com/magazine\\_satisfaction.php](http://www.programmez.com/magazine_satisfaction.php)

**PRO** le magazine du développement **grammez!**

numéro 123 - octobre 2009

# sommaire

\\ actus	
L'actualité en bref .....	6
Agenda .....	6
\\ sgbd	
Les moteurs de stockage dans MySQL .....	11
\\ gros plan	
HTML 5 : la révolution ? .....	14
\\ webmaster	
Comment tester vos pages web ? .....	22
\\ carrière	
Comment les développeurs gèrent-ils la crise ? .....	27
\\ dossier :	
<b>WINDOWS 7 :</b>	
<b>le guide du développeur</b> (1re partie)	
la compatibilité des applications .....	34
Découvrez le Windows API Code Pack .....	38
Utiliser le multitouch dans ses applications .....	45
Ecrire des services déclenchés .....	49
Intégrez la biométrie à vos applications avec Windows 7 .....	51
La migration vers Windows 7 : l'avis d'un développeur .....	54
\\ développement web	
GWT : comment ça marche .....	56
La sécurité dans Spring .....	59
\\ code	
.Net Micro Framework : l'autre visage de .Net ! .....	62
Intégration de OSWorkflow avec les frameworks Spring IoC et Acegi .....	65
A la découverte du modèle de composants Fractal .....	67
Ecrire un modèle d'extension pour Drupal .....	70
Tracez et hackez les processus sous Linux .....	74
Intégrer une webcam en Flash-PHP côté serveur .....	79
\\ temps libre	
Les livres du mois .....	82



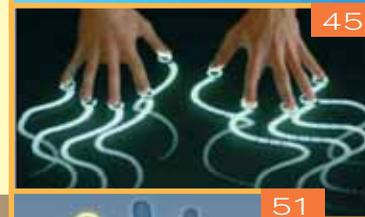
8



27



32



45



51



79

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.



*«Intel® Parallel Studio accélère la mise sur le marché, au niveau mondial, de nos logiciels.»*

ERIC ROSIER, VICE-PRÉSIDENT, INGÉNIERIE, ENVIVIO



Le parallélisme n'est pas uniquement destiné au calcul intensif. Développez de riches applications de bureau et nomades.

**Créez un code évolutif.**

Version d'évaluation gratuite: [www.intel.com/go/parallel](http://www.intel.com/go/parallel)



## L'utopie du langage

Interopérabilité, joli terme, omniprésent dans l'informatique : bureautique, SOA, web service, gestion des identités, vidéo, audio, 3D... Il y a tout de même un endroit où l'interopérabilité manque encore d'adhérence : les langages de programmation. Oh bien sûr, on nous gratifie de compatibilités et d'intégration plus ou moins complètes : JRuby pour Ruby et Java, IronPython pour .Net et Python, Mono pour un .Net non Microsoft, un clone de Java ici, un Java dans App Engine là-bas. Et nous pourrions multiplier les exemples.

C'est de bonne guerre. Java sort une nouvelle fonction super sympa, les autres suivent avec plus ou moins de bonheur. Même dans .Net, VB et C# n'étaient pas au même niveau fonctionnel... jusqu'à l'an dernier. La course aux nouveautés dans les langages pose tout de même un problème que les éditeurs et communautés reconnaissent : tout cela alourdit le langage, la syntaxe et le code. Nécessite de la formation, des adaptations parfois. Heureusement, cette course effrénée se ralentit quelque peu. Pour contrer la complexité d'un langage, on nous en invente un nouveau, 'plus mieux' que l'ancien. Ruby est né parce que son inventeur ne trouvait pas de langage répondant à ses besoins. Ou comment s'amuser quand on n'a rien à faire un samedi soir... d'hiver de préférence.

A force d'empiler, il faudra bien envoyer en cure sévère notre langage préféré. Prenez la JDK Java, pour optimiser un peu, on la découpe. Manière, aussi, de cacher, l'embonpoint de la bête... Mais alors que faire ? Avec les nouveautés systèmes et matérielles, il faut bien adapter, compléter les langages pour aider nos amis développeurs. Les bibliothèques parallèles servent à cela. HTML est trop pauvre. Pas de problème, on nous pond HTML 5 qui pour arriver à un consensus laborieux perd des pans fonctionnels et son implémentation partielle varie d'un navigateur à un autre. Nous pourrions en faire un inventaire à la Prévert. Et que dire des langages concurrents, qui font la même chose ou quasiment : Flex, Silverlight, JavaFX... ? Finalement, le boulier avait son charme.

On nous parle sans cesse de standards, de normes. Peut-être faudra-t-il faire un peu le ménage ! Les langages normalisés ISO existent: Fortran, C++, Cobol, Pascal, ECMAScript, Ada. La question est de savoir ce que nous voulons faire et comment le faire. Il paraît difficile de supprimer 90 % des langages, pour ne garder que certains langages étalon. Je vois déjà les pugilats.

Par contre, il serait possible, même si cela paraît follement naïf de notre part, de définir des extensions, des fonctions communes à tout le monde, opérant et implémentées de la même manière. Cela éviterait bien des soucis aux développeurs. Et pourquoi ne pas rêver à une véritable interopérabilité des langages ? Utiliser n'importe quel langage avec n'importe quel autre dans quelque IDE ou éditeur que ce soit ! Là, on fera un immense pas en avant.

Les enjeux sont colossaux, plusieurs dizaines de milliards d'euros. Est-ce utopique que de rêver à un « core language » sur lequel on grefferait des plug-in, des composants provenant des éditeurs, des développeurs, des communautés ? Dans une informatique de plus en plus complexe, faut-il chercher à l'alourdir encore et encore ? Voilà la question à laquelle nous devons répondre d'ici à 5 ans.

■ François Tonic  
Rédacteur en chef

Rédaction : redaction@programmez.com  
 Directeur de la Rédaction : Jean Kaminsky  
 Rédacteur en Chef :  
 François Tonic - ftonic@programmez.com  
 Ont collaboré à ce numéro : F. Mazué, C. Remy, P. Borghino  
 Experts : O. Dasini, A. Gadal, F. Goldgewicht, N. Franchet, T. Jaskula, S. Crozatier, P. Manac'h, L. Vo Van, T. Laurens, B. Chauvin, J. Dollon, F. Queudret, F. Hamel, A. Ancelin, P. de Saint Staban, M. Najjar, A. Sousak, M. Isvy, F. Santin, J. Corioland, Y. Davin, D. Doussot,  
 Illustration couverture : Microsoft, Logitech  
 Publicité : Régie publicitaire, K-Now sarl  
 Pour la publicité uniquement :  
 Tél. : 01 41 77 16 03 - diff@programmez.com  
 Editeur : Go-O2 sarl, 21 rue de Fécamp 75012 Paris - diff@programmez.com  
 Dépôt légal : à parution - Commission paritaire : 0712K78366 ISSN : 1627-0908  
 Imprimeur : ETC - 76198 Yvetot  
 Directeur de la publication : J-C Vaudecrane  
 Ce numéro comporte un encart Component Source sur les abonnés et une partie du tirage.

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

L'INFO  
PERMANENTE  
WWW.PROGRAMMEZ.COM



PROCHAIN  
NUMÉRO

N°124 - novembre 2009,  
parution 31 octobre

✓ **Dossier spécial !**  
Guide du  
développeur  
Windows 7  
2e partie

✓ **RIA**  
La guerre des plates-formes  
choisir entre JavaFX,  
Silverlight, Flex !

✓ **Enquête**  
Choisir son langage  
de développement

■ Le projet Mono vient de dévoiler **MonoTouch**. Cette édition spéciale de Mono (pile .Net open source) se dédie à l'iPhone d'Apple. Cette solution est une alternative aux outils officiels Apple et au langage maison Objective-C. MonoTouch permet de créer des applications mobiles C# tout en supportant les API du mobile pommé. Comme pour le SDK Apple, il faut disposer d'un Mac pour le développement, du SDK et d'un terminal et bien entendu d'un compte développeur iPhone. Une version d'évaluation est disponible. Les tarifs vont de 399 à 3999 dollars US. Site : <http://monotouch.net>

■ WebKit annonce l'intégration de WebGL dans le projet WebKit, le moteur de rendu utilisé par Android, Nokia et Apple. **WebGL** est issu du groupe Khronos. WebGL est une librairie 3D pour accélérer la 3D en utilisant la puce graphique. La technologie s'appuie sur OpenGL ES 2 et Javascript. Surtout, son implémentation dans Webkit signifie l'absence d'un module externe complémentaire. Les premiers navigateurs compatibles ne devraient pas apparaître avant les premiers mois de 2010.

■ Le développement mobile est toujours le parent pauvre du développeur. **Sybase** et **IBM** s'associent pour promouvoir plus largement le commerce mobile. Les deux éditeurs utiliseront leurs solutions et forces commerciales pour favoriser le commerce mobile et les solutions gravitant autour. Avec l'explosion des Smartphones, c'est un marché immense qui s'ouvre enfin !

## SDL Deux nouveaux outils pour développer sécurisé

Microsoft vient d'ajouter deux nouveaux outils à son offre SDL (développement sécurisé) : **BinScope Binary Analyser** et **MiniFuzz File Fuzzer Tool**. **BinScope Binary Analyser** servira aux développeurs et testeurs pour analyser leurs binaires. Il vérifie la présence des mesures de sécurité et leur bonne « activation ». Au final, l'outil permet de dire si une application est potentiellement vulnérable par rapport aux principales failles de sécurité connues. Il fonctionne aussi bien pour les binaires 32 que 64-bit et avec les assemblées managées (.Net). Il ne supporte pas Windows CE et le Mac. Il possède trois étapes : exécution de l'assistant, chargement des résultats dans Switrack puis passage en revue des problèmes. L'interface a été simplifiée au maximum : une fenêtre, 4 onglets ! Détail important,



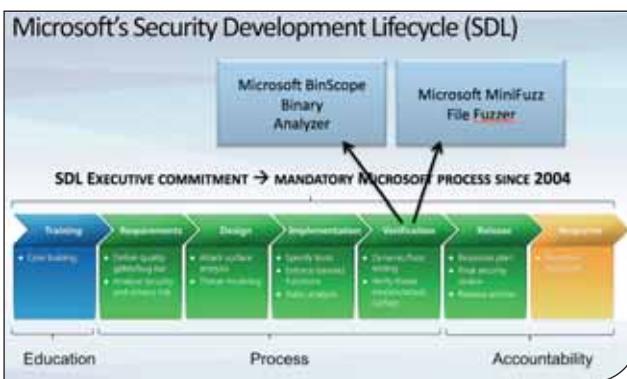
**BinScope** suit la même logique qu'un debugger pour les symboles. Il peut scanner les différents types de linker (au niveau compilateur) : GS, safeseh, nxcompat, ASLR, APTCA. Attention tout de même car sur certains linkers, il y aura des limitations de vérifications SDL. **MiniFuzz File Fuzzer Tool** permet de vérifier la sécurité non plus au niveau binaire mais le

code source. Outil automatique, il détecte et identifie les failles, les trous de sécurité du code et fournit une analyse sur le risque potentiel encouru. **MiniFuzz** est un outil en C++.

Le développeur trouvera plus rapidement avec **MiniFuzz** les corruptions dans les sorties et problèmes inattendus.

Là aussi l'interface se veut simple et rapide à utiliser. Il s'adapte au niveau du développeur et s'intègre à Team Foundation Server.

Il peut s'utiliser seul ou dans Visual Studio (tout comme **BinScope**). A cela s'ajoute un manuel d'intégration du SDL Process Template, disponible depuis mai dernier. Car la méthode et les outils SDL ne sont pas toujours bien compris par le développeur.



## agenda \

### OCTOBRE

- Du 1er au 02 octobre 2009, Paris 8e, EUROSITES GEORGE V - 28, avenue George V. **Open World Forum 2e édition** Le 1er forum mondial du Logiciel Libre [www.openworldforum.org](http://www.openworldforum.org)
- Du 06 au 08 Octobre, Paris Porte de Versailles **IP Convergence**, le salon des équipements, services et applications de communication IP <http://www.ipconvergence.fr/>
- Du 06 au 7 octobre 2009, Paris **Microsoft Days 2009 / Rencontres techniques**

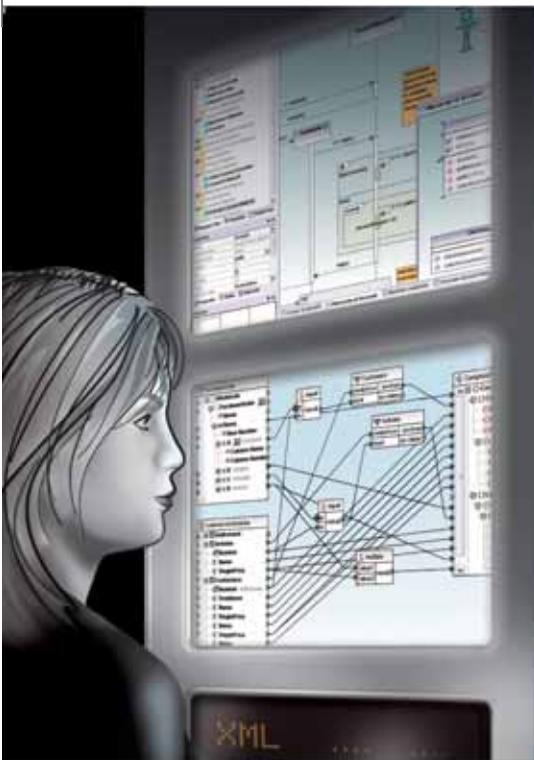
Pour booster vos compétences avec plus de 20 sessions techniques, le 15/10 2009 à Lille, le 19/10 à Marseille, le 20/10 à Strasbourg. <http://msdn.microsoft.com/fr-fr/microsoft-days.aspx#>

- Du 06 au 08 octobre, CNIT Paris La Défense, - **Serveurs et Applications**, le seul salon réellement dédié au System i <http://www.salons-solutions.com/fr/as400>

- **Solutions BPM 2009**, le Salon des Solutions de Pilotage des Activités Métiers de l'Entreprise. <http://www.groupe-solutions.fr>

- Du 07 au 10 octobre, Monaco, 9e édition des **Assises de la sécurité et des systèmes d'information**, Le rendez-vous annuel incontournable de la communauté sécurité. <http://www.lesassisesdelasecurite.com>

- 8 octobre - Université de Strasbourg, UFR de Math info. **DeveloperForum5** : une journée orientée développeur, nouvelles technologies. Le thème de cette édition : le futur du web. site : <http://www.developerforum.eu/>



Offrez-vous  
toutes les meilleures  
technologies à un prix  
abordable grâce à la suite  
complète des outils d'intégration  
professionnels d'Altova®



Découvrez comment le MissionKit® d'Altova, la suite intégrée d'outils XML, UML et de bases de données, peut vous permettre de mettre à profit les technologies et logiciels professionnels existants, tout en intégrant les technologies modernes, et ce sans vous ruiner.



Le MissionKit d'Altova comprend plusieurs outils intelligents pour l'intégration professionnelle:

**MapForce®** – l'outil graphique de mappage de données

- Conversion des données par glisser-déposer, plus transformation instantanée et génération de code
- Prise en charge du mappage des données XML, de bases de données, EDI, XBRL, Excel® 2007, des fichiers texte et des services web

**DatabaseSpy®** – l'outil de requête et de conception multi-bases de données

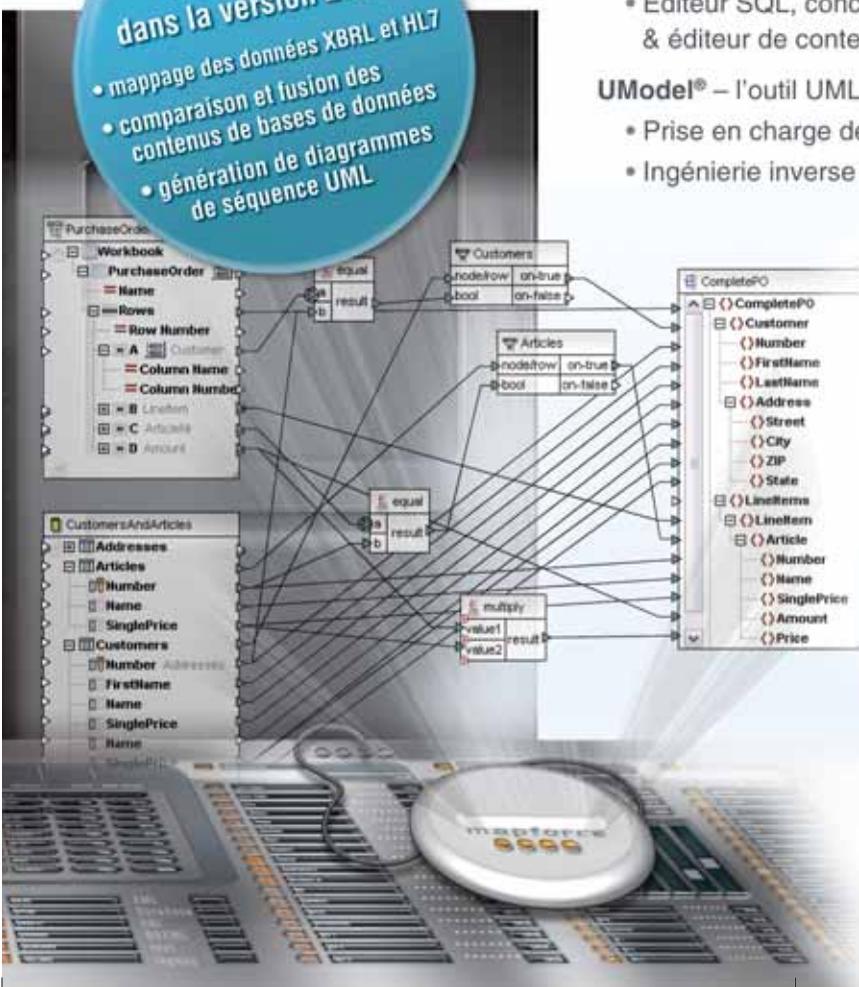
- Prise en charge de toutes les principales bases de données relationnelles à l'aide d'un outil unique
- Éditeur SQL, concepteur graphique de bases de données & éditeur de contenu

**UModel®** – l'outil UML pour la modélisation de logiciels

- Prise en charge des 14 types de diagrammes UML et du BPMN
- Ingénierie inverse et génération de code en Java, C#, VB.NET

Nouveautés  
dans la version 2009:

- mappage des données XBRL et HL7
- comparaison et fusion des contenus de bases de données
- génération de diagrammes de séquence UML



↓ Téléchargement gratuit!

Testez avant d'acheter avec une version d'essai de 30 jours gratuite et entièrement fonctionnelle, disponible sur [www.altova.com](http://www.altova.com).

## actu \ en bref

■ Mozilla lance **Jetpack**. Il s'agit d'une API pour développer des add-on Firefox utilisant les technologies web. Jetpack peut servir à de nombreuses applications : enregistreur, notification mail... Le but est de simplifier la conception d'add-on Firefox.

Site : <https://jetpack.mozillalabs.com>

■ Une version Android de **JRuby** est en cours de développement. Pour rappel, JRuby permet d'intégrer dans les applications Java, du code Ruby. C'est une bonne nouvelle pour la communauté. On devrait en savoir plus lors de la JRubyConf fin novembre.

■ A peine **Silverlight 3** disponible, Microsoft commence à parler de la version 4.0 qui sera livrée à une date inconnue. Le futur Silverlight supportera la multidiffusion, la gestion déconnectée des DRM. Silverlight est déjà capable de diffusion en Haute Définition de la vidéo en direct, fonction qui devrait se renforcer avec la v4.

■ **MonoDevelop 2.2** est disponible en pré-version. Cette version supporte officiellement Windows et MacOS X ou encore le multicible au niveau runtime. L'éditeur de code s'enrichit de nouvelles fonctions : sauvegarde automatique, nouveau moteur de refactoring, un debug plus performant, ajout d'un add-on ASP.Net MVS.

Site : <http://monodevelop.com>

## Cloud Google livre App Engine SDK 1.2.5

Doucement mais sûrement, Google consolide et complète App Engine. La version 1.2.5 apporte des nouveautés très intéressantes pour le développeur. La principale est l'apparition de XMPP (Jabber). Il s'agit d'un standard ouvert de messagerie instantanée et un des plus populaires. Ce support est disponible en Python et Java comme App Engine dans les deux langages. Il sera ainsi possible de construire des applications de communication dans le cloud tout en supportant les flux Jabber des applications tiers ou venant de Google Wave et Google Talk. L'autre nouveauté bienvenue est l'arrivée de l'API Task Queue pour Java. Jusqu'à présent, seul Python supportait



cette fonction de tâches délayées dans les applications App Engine. Désormais Java est au même niveau. Autre bonne nouvelle, le quota autorisé de la Task Queue est fixé à 100 000. Le but de l'équipe Google est de supprimer tout quota dans le futur. Autre nouveauté, une fenêtre de lancement pour Windows que les utilisateurs MacOS X possédaient déjà ! Pour en savoir plus : <http://googleappengine.blogspot.com>

## Cloud VMware rachète SpringSource qui rachète CloudFoundry

Le cloud computing est plus que jamais le terrain de jeux des éditeurs. VMware tente de rester dans le coup en étendant son offre. Le rachat de SpringSource permet de se constituer une plate-forme applicative dans le cloud. Dans le même temps SpringSource rachetait CloudFoundry dont le but est de proposer une plate-forme pour le déploiement et l'exécution pour les applications Java. Le tout respectant le cycle de l'application Java : build, exécution, administration. Foundry est donc une

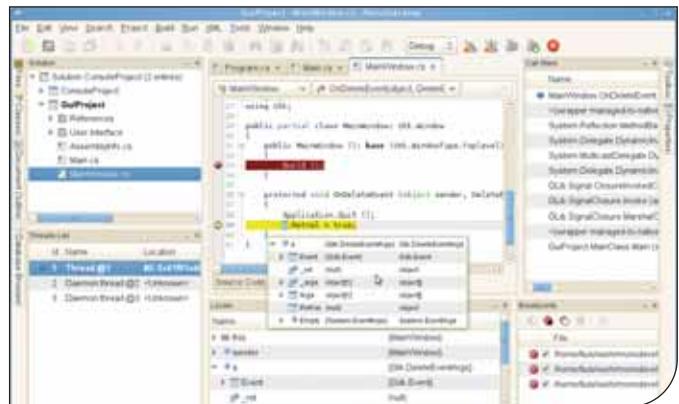


plate-forme de cycle de vie des applications Java, Spring et Grail. Le tout fonctionne sur Amazon EC2. CloudFoundry s'appuie sur Cloud Tools. Cloud Tools est une suite d'outils open source pour déployer, manager et tester les applications Java EE dans un contexte Amazon EC 2. Cette suite se compose de trois éléments : Amazon Machine Images, EC2 Deploy et Maven / Grails.

Cloud Tools supporte aussi Elastic Block Store. EBS fournit les blocs des volumes de stockage utilisés dans les instances EC2. Décidément, gérer les instances et déploiements EC2 devient un sport à la mode car ce ne sont pas les premiers à le faire. Reste maintenant à étendre Cloud Tools aux autres infrastructures cloud...

Site : <http://www.cloudfoundry.com>

## Système \ Apple libère Grand Central Dispatch

Avec la sortie de Snow Leopard, MacOS X 10.6 fin août, plusieurs technologies liées à une meilleure charge processeur ont été dévoilées : OpenCL, Grand Central. Grand Central permet de diriger les tâches dans les cœurs du ou des processeurs afin d'optimiser leur utilisation pour une répartition plus homo-

gène. Si les gains ne sont pas forcément énormes, Grand Central allège les charges processeurs et évite une saturation excessive. Apple n'a pas toujours été bien vu des communautés Open Source mais l'annonce de libdispatch améliorera les choses. Libdispatch est l'ouverture du code source de

Grand Central permettant d'améliorer et d'implémenter cette technologie dans ses applications. Le projet est mis en licence Apache 2.0. Pour le moment, cette librairie concerne MacOS X mais rien n'interdit un futur portage dans d'autres Unix et Linux. A vous de jouer ! site : <http://libdispatch.macosforge.org/>

# ENTREZ DANS UN DATACENTER DE NOUVELLE GENERATION

**SELON GARTNER 50% DES CENTRES DE DONNÉES NE SERONT PLUS EN MESURE DE RÉPONDRE AUX EXIGENCES D'ALIMENTATION ÉLECTRIQUE ET DE DISSIPATION THERMIQUE EN 2008 ! \***

... pas celui d'ASPSERVEUR

◀◀ ASPERVEUR CRÉE LE CENTRE DE DONNÉES ULTIME À LA CIOTAT (13) ▶▶▶



## 8 X plus d'énergie, 8 X plus de dissipation



- ▶ Capacités en énergie 8 fois supérieures à celles des Datacenters traditionnels (jusqu'à 16 kW/baie)
- ▶ Dissipation thermique 8 fois supérieure à celle des Datacenters traditionnels (jusqu'à 16 kW/baie)
- ▶ Bâtiment de 3800 m2 (permet d'héberger autant de serveurs qu'un Datacenter traditionnel de 14 400 m2 !)
- ▶ Economie d'énergie proche de 40% grâce aux onduleurs à haut rendement et aux climatiseurs de précision
- ▶ Parfaite redondance des équipements, très haute disponibilité garantie par contrat SLA : 99.995% (Tier 4)
- ▶ Sécurité ultime (barrières infrarouges, radars linéaires, cameras, lecteurs de badges sectorisés, baies à serrures électromagnétiques avec accès par badge, détecteurs de choc, zone en sismicité 0, brumisateurs, système d'extinction incendie SIEMENS par azote, coffre fort pour les backups, réseau privé de rétention et d'écoulement des eaux pluviales, deux groupes électrogènes...)
- ▶ Agréments sécurité, assurances et code du travail
- ▶ Connectivité maximum : double adduction fibres optiques DWDM (jusqu'à 2 x 83 Gbps)
- ▶ Présence des principaux opérateurs (ORANGE BUSINESS, SFR NEUF CEGETEL, COGENT NETWORKS, ASPERVEUR ...)
- ▶ Attribution d'adresses IP (membre du RIPE)
- ▶ Emplacements en toiture pour antennes GSM/WIFI/WIMAX/PARABOLES/LASER
- ▶ Services 24H/24 : Eyes & Hands, ingénierie CISCO (Selected Certified Partner), infogérance, monitoring et alarmes, spare
- ▶ Collocation, Housing, location de serveurs, location de baies, location de suites sur mesure, bureaux, stockage, coffre-fort
- ▶ Service de backup IBM TIVOLI / SAS EQUALOGIC en cluster sur deux datacenters
- ▶ Service de load-balancing CISCO CSS et cluster sur deux Datacenters (Marseille - La Ciotat)
- ▶ Possibilité de PRA et données synchrones avec le Datacenter de Marseille (PING >1 ms, 2 X 41 kms de fibres optiques)
- ▶ Conseil en architectures de haute disponibilité (tolérance de panne, cluster, SAN/SAS, PRA, accélération TCP/IP, sites à forte charge)

\* Source : Magazine "L'informaticien" mai 2007, dossier Hébergement

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

N° indigo : 0811 651 583



## actu \ en bref

■ **Kapitec Software** a signé un partenariat stratégique avec l'éditeur Frologic. Kapitec devient le distributeur pour le marché français de Frologic. Cet éditeur permet de tester les applications SWT, Eclipse RCP, AWT, Swing, Qt, Carbon, Cocoa, HTML et Ajax. Il se démarque des autres outils en permettant d'écrire ses propres tests. Site : [kapitec.com](http://kapitec.com)

■ **Infragistics**, annonce la sortie de **NetAdvantage for .NET 2009 Volume 2** : Win Client, un ensemble d'outils de développement WPF et



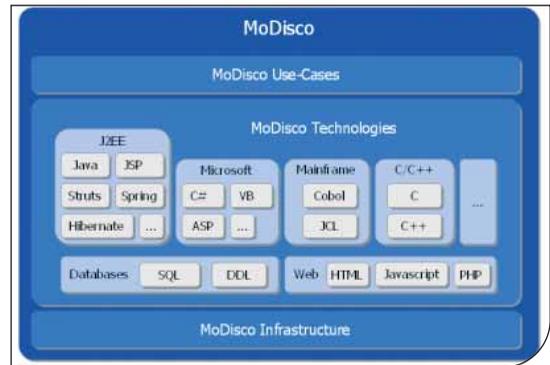
Windows Form. Ce volume 2 introduit notamment des composants Data Grid que l'éditeur présente comme les plus rapides du marché, et capables de traiter efficacement des millions de lignes de données.

■ **Altova** ajoute une formation **XBRL** dans son programme en ligne Online Training. Il consiste en 5 modules d'apprentissage incluant des tutoriaux détaillés que le développeur, peut utiliser dans MissionKit 2009. Il permet notamment de créer des extensions de taxinomies dans XMLSpy. L'éditeur est très sensible au XBRL et à son apprentissage. Site : [www.altova.com](http://www.altova.com)

■ L'un des outils phares de Microsoft, **Project**, arrive en version 2010. L'outil s'insère dans une approche plus globale avec Sharepoint 2010, Exchange 2010 et Office 2010. Cette solution sera disponible en trois éditions : standard, professionnelle et serveur. L'éditeur met en avant une gestion centralisée des portfolios de projets et des projets notamment par une nouvelle interface, des flux personnalisables. La gestion des demandes a été largement revue.

## Modèle Mia-Software supporte Eclipse Galileo

Avec chaque nouvelle version d'Eclipse, les outils se mettent à jour. Mia-Software, éditeur français de solutions de modélisation, annonce pour MoDisco, dédié à la modernisation des applications, la compatibilité avec Eclipse Galileo. Cette nouvelle version, v0.7, est un sous-projet d'Eclipse. Le but de cette solution est de reconstruire les modèles automatiquement à partir d'une application, d'où le nom MoDisco, pour Model Discovery. Le but de l'éditeur est de créer une véritable plate-forme



d'outils de modernisation et notamment pour les anciennes applications Java dont les premières versions rentrent dans la catégorie patrimoniale. Le modèle généré s'appuie sur UML mais l'ouverture de la solution permettra dans le futur de rajouter des fonctions spécifiques. Les fonctions de base sont : l'analyse de la qualité, la compréhension de l'application, la migration et le refactoring. La prochaine version sera disponible à la fin de l'année.

Site : [www.eclipse.org/gmt/modisco/](http://www.eclipse.org/gmt/modisco/)

## SGBD

### Oracle dégage la release de Oracle 11g

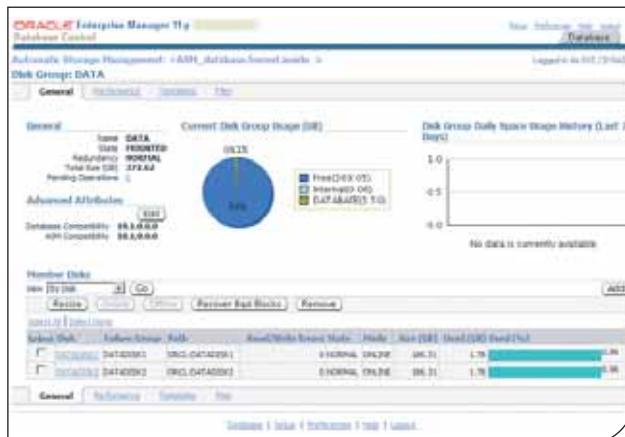
Le leader des bases de données pour cette mise à jour majeure de son produit phare est simple et direct : réduction des coûts serveur par 5, réduction de besoins en stockage, amélioration des performances en mission critique par 10, amélioration de l'administration, simplification du portefeuille logiciels IT. Si nous prenons la partie stockage, les équipes d'Oracle ont travaillé sur différents éléments comme l'*Automatic Storage Management* qui permet de mirroring les bases sans ajout d'un outil tiers. Et de nouveaux

stockages sont rajoutés à la volée dans l'environnement de données. Le tout est couplé à un système de fichier spécifique le *Automatic Storage Management File System*. Autre point d'amélioration, le partitionnement et le coût global du stockage. Ainsi pour faciliter les traitements (et donc les performances), les tables les plus grosses sont segmentées en plus petites. Il s'agit aussi d'optimiser le stockage utile (le stockage réellement utilisé) et d'éviter le gaspillage de place. Enfin, la compression est un autre mécanisme

important de cette partie du SGBD. La disponibilité est un des nombreux objectifs de cette r2 dévoilée officiellement fin septembre.

Pour Oracle cela passe par le concept *Maximum Availability Architecture* qui doit utiliser au mieux les éléments de redondance des serveurs. Il s'agit en particulier d'ajouter automatiquement de toutes nouvelles ressources dans le plan de disponibilité. Sur la partie développement, Oracle 11g r2 continue à supporter le plus de langages possibles : Java (support historique), .Net avec Oracle Developer Tools pour Visual Studio (incluant le support du déploiement .Net, de VB et C#), PHP, Oracle Call Interface pour la partie C, C++, Fortran et Cobol, le langage maison PL/SQL. Côté outils, on dispose toujours d'Oracle Application Express, de SQL Developer et de SQL Developer Data Modeler.

Site : [www.oracle.com](http://www.oracle.com)



# Les moteurs de stockage dans MySQL

Une des particularités de MySQL est son architecture « pluggable storage engine » qui permet de choisir la manière dont le serveur stocke et traite les données.

En effet, à chaque table est associé un moteur de stockage pouvant implémenter (ou non) des fonctionnalités telles que les transactions, les clés étrangères, les algorithmes d'index hash ou fulltext... Chaque moteur ayant des caractéristiques propres, nous vous conseillons de bien les étudier pour en connaître les avantages, les limites et surtout comment les configurer.

## Les moteurs de votre serveur

Il est facile de connaître les moteurs présents en utilisant la base « INFORMATION SCHEMA » qui stocke les méta données du serveur.  
mysql> SELECT ENGINE,SUPPORT,COMMENT FROM information \_schema.ENGINES IG

## Ajout dynamique d'un moteur de stockage

Depuis MySQL 5, il n'est plus nécessaire de (re)compiler le code source du serveur pour ajouter un nouveau moteur, il suffit de copier une librairie dans le répertoire prévu à cet effet et de la charger sans redémarrer le serveur.

Si nous prenons pour exemple le moteur PBXT :

```
INSTALL PLUGIN SONAME
```

Nous pouvons le supprimer de la même façon: UNINSTALL PLUGIN ;

Attention : si vous supprimez un moteur de stockage utilisé par une ou plusieurs tables, les données seront toujours présentes mais vous ne pourrez pas y accéder. L'installation d'un nouveau moteur doit être réalisée par le DBA. Cette opération nécessite en effet des droits spécifiques pour être exécutée correctement. Il s'agit ici notamment du droit d'écriture dans la table mysql.plugin.

## Comment choisir son moteur ?

Les paramètres à prendre en considération reposent essentiellement sur les besoins de vos applications.

Par exemple si vous avez besoin de transactions, pour être plus résistant aux pannes, pour avoir une meilleure concurrence, vous allez vous orienter vers un moteur transactionnel comme InnoDB, Falcon, PBXT,... Utiliser un moteur transactionnel signifie aussi des besoins plus grands en ressources car davantage d'écritures sont nécessaires sur disque.

Voici quelques uns des principaux moteurs:

**MyISAM** : le moteur par défaut de MySQL. Il n'est pas fourni sous la forme d'une librairie et nous ne pouvons pas le supprimer. Presque toutes les tables systèmes l'utilisent. Il pose des verrous au niveau de

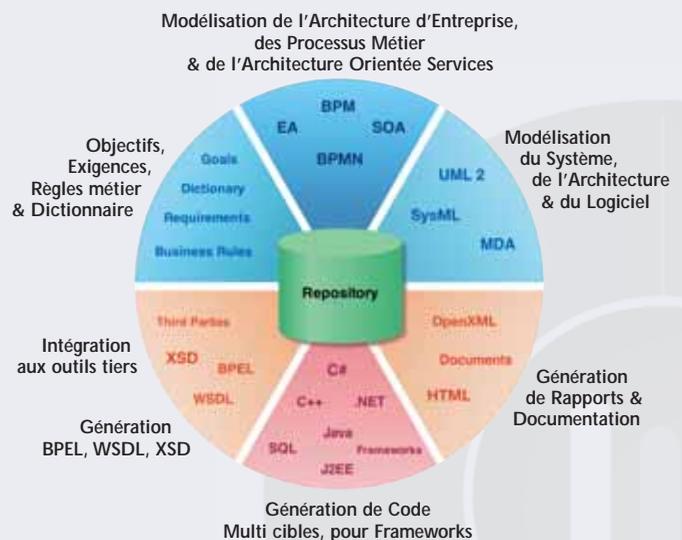
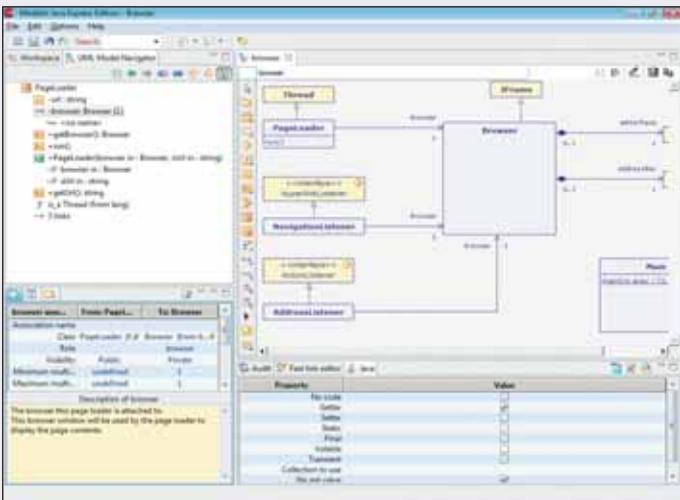


## Modelio : une nouvelle génération d'outil

*Modéliser n'a jamais été aussi simple et productif !*

### Modelio: une offre de modélisation unique !

- Ergonomie simple, productive et familière aux développeurs (RCP/Eclipse)
- Modélisation intégrée de UML2, BPMN, SysML, l'Architecture d'Entreprise, les exigences, le dictionnaire, ... dans un seul référentiel
- Travail de groupe distribué, intégré à SVN/Subversion
- Génération Java, C#, C++, SQL, XML, XSD, BPEL, WSDL, Hibernate...
- MDA simple et puissant - transformation, extensibilité et adaptabilité



### Modelio est disponible en trois éditions

- **Free** : Un outil complet de modélisation gratuit !
- **Express Java** : Un outil de développement UML2/Java performant pour seulement 100 € !
- **Enterprise** : La solution de modélisation complète, supportant le travail de groupe, extensible avec une riche palette de modules de modélisation et de génération disponibles sur étagère



Téléchargez la nouvelle version de Modelio !  
[www.modeliosoft.com](http://www.modeliosoft.com)

sales@modeliosoft.com  
Tél. : 01 30 12 18 40

## sgbd \\\ moteur

la table, ne supporte pas les transactions mais possède des fonctionnalités intéressantes comme les recherches de textes (FULL TEXT SEARCH) et les index géographiques (GIS). Il est encore utilisé pour les applications avec peu de concurrence et des volumes raisonnables.

**MARIA** : à l'origine MARIA devait être une évolution de MyISAM qui corrige les problèmes de corruptions qui apparaissent assez souvent après un crash. Comme ISAM a laissé sa place à MyISAM, nous nous attendions à ce que MARIA remplace MyISAM. Si vous suivez l'actualité de MySQL, Maria est maintenant le moteur par défaut du projet MariaDB, une version « alternative » de MySQL. MARIA supporte les transactions, pose des verrous au niveau des lignes. Pour supporter les transactions il utilise MVCC (Multiversion concurrency control). La plupart des moteurs supportant les transactions dans le monde MySQL utilisent le même algorithme. Le MVCC permet de proposer à chaque utilisateur connecté à MySQL un cliché de la base. Tous les changements effectués par l'utilisateur en question seront invisibles des autres utilisateurs tant que la transaction n'aura pas été validée (commit).

**InnoDB** : Un des moteurs historique de MySQL, le premier à supporter les transactions ACID (Atomique, Consistant, Isolation, Durabilité). Depuis 2008, le moteur InnoDB est disponible sous 2 formats : livré avec MySQL ou sous la forme d'une librairie. Il appartient à InnoDB/Oracle. Depuis la version 5.1.38, le plugin est livré par défaut. C'est de nos jours le moteur le plus utilisé. <http://www.innodb.com/>

**XtraDB** : Une évolution du plugin InnoDB développé par Percona (société de consulting spécialisée dans la performance) qui a rajouté des fonctionnalités et des patches pour ses clients. Il est disponible sur <http://www.percona.com/mysql/xtradb/>

**FALCON** : disponible avec la 6.0, le moteur a été créé par MySQL AB en vue de supporter les transactions et avec pour objectif de monter en charge linéairement. Depuis le rachat par Oracle beaucoup d'interrogations subsistent quant au devenir de ce moteur.

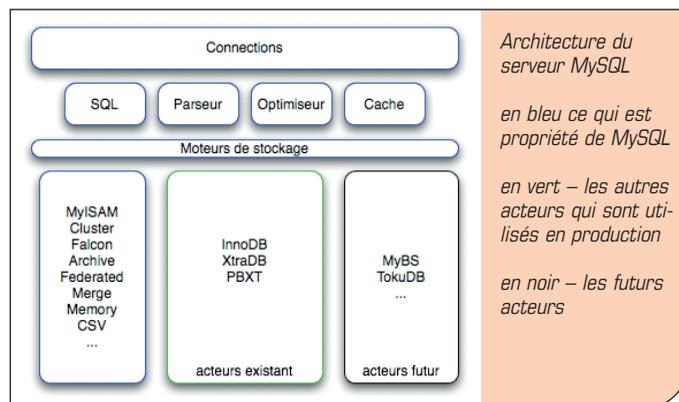
**MEMORY** : anciennement connu sous le nom de HEAP, les données et les index sont stockés en mémoire vive ce qui rend l'accès très rapide. Comme elles sont en mémoire vous perdez le contenu lors d'un redémarrage de l'instance de MySQL, mais pas sa structure.

**ARCHIVE** : les données sont stockées sous une forme compressée. Les seuls types de requêtes supportés sont SELECT et INSERT. Par contre le gain de place sur disque est important. C'est un moteur idéal pour archiver des données.

**FEDERATED** : permet d'accéder à des données situées dans une autre instance de MySQL sur un serveur local ou distant.

**CSV** : ce moteur permet de stocker les informations sous la forme csv (Comma Separated Value), un fichier texte où les valeurs sont séparées par une virgule. Il permet de faciliter les échanges vers d'autres applications.

**Blackhole** : une table utilisant blackhole (trou noir) ne stocke absolument rien. Ce moteur peut être utilisé pour valider des requêtes SQL ou pour distribuer les données sur différents serveurs grâce à un filtrage opéré au travers de la répllication.



Architecture du serveur MySQL

en bleu ce qui est propriété de MySQL

en vert – les autres acteurs qui sont utilisés en production

en noir – les futurs acteurs

**Exemple** : ce moteur ne fait rien. Il sert d'exemple aux développeurs pour implémenter leur propre moteur de stockage.

**NDB** (Network Database) : utilisé par MySQL Cluster essentiellement et ne peut donc pas être utilisé séparément.

### Les autres moteurs

D'autres moteurs sont disponibles qu'ils soient open source communautaires ou développés par des sociétés propriétaires. Ces moteurs ne sont pas supportés officiellement par MySQL et sont donc à utiliser à vos risques et périls. L'organisme à l'origine du développement s'occupe du support, de la documentation, et des bugs. La plupart des moteurs sont référencés sur MySQL Forge : <http://forge.mysql.com/>

**PBXT** : Un des moteurs les plus en vue en ce moment est PrimeBase XT. Il a été conçu pour les environnements web à forte concurrence, et utilise une architecture basée sur l'écriture de fichiers de logs. <http://www.primebase.org>

**BLOB Streaming Engine (MyBS)** : MyBS devait être à l'origine une évolution à PBXT. Les deux moteurs étaient développés en parallèle par Primebase. L'architecture de MyBS a changé, elle transforme MySQL en un serveur de médias capable de diffuser (streaming) images, films, mp3... et autres objets binaires (BLOBS) à partir de la base ou vers la base. <http://blobstreaming.org/>

**SolidDB** pour MySQL est un moteur fourni par Solid et basé sur leur propre base de données SolidDB. Solid a été racheté par IBM. <http://www-01.ibm.com/software/data/soliddb/>  
<http://sourceforge.net/projects/soliddb/>

**TokuDB** : un nouveau venu qui incorpore la technologie fractale de Tokutek. Les performances affichées sont assez bluffantes. C'est un moteur à suivre. <http://tokutek.com>

■ **Pascal Borghino**  
Architecte base de données et auteur sur [DBNewz.com](http://DBNewz.com)

■ **Olivier Dasini**  
Expert MySQL chez Orange Business Services - et auteur sur [dasini.net](http://dasini.net)

■ **Arnaud Gadal**  
DBA chez Virgin Mobile - et auteur sur [DBNewz.com](http://DBNewz.com). Un livre est en préparation sur ce thème.



# Introduisant NetAdvantage pour .NET 2009 Volume 1

Des composants interface utilisateurs supérieurs pour Windows Forms, WPF, ASP.NET et Silverlight

Pour de plus amples infos : [infragistics.com](http://infragistics.com)



Four Platforms. One Package.

N° Vert 0800 667 307

grids

tree

menus

navigation

charts

& more!

# HTML 5 : la révolution ?

La spécification HTML 5 fait de plus en plus parler d'elle : au cours de ces 10 ans de HTML 4, le Web a évolué et de nouveaux besoins sont apparus. En particulier, concernant la notion d'application Web : les lacunes de HTML 4 en la matière ont été comblées de diverses façons par les différents frameworks apparus mais ce manque de solutions natives a grandement complexifié la programmation Web.

La spécification HTML 5 apporte de nombreuses améliorations au langage HTML, en termes de flexibilité, d'interopérabilité et de richesse de l'expérience utilisateur. A un tel point qu'il est souvent élevé au rang de technologie RIA « killer de Flex, Silverlight et JavaFX ».

La pauvreté de HTML en termes d'ergonomie et d'interaction utilisateur ont conduit les éditeurs à proposer des solutions permettant d'enrichir les interfaces utilisateur de ces applications : depuis 2002, des dizaines de frameworks permettant de concevoir des **Rich Internet Applications** (RIA) ont vu le jour, contribuant à l'élaboration de ce qui a été nommé « Web 2.0 ». Parmi ces solutions, on trouve des bibliothèques AJAX (HTML/JavaScript/CSS) et des frameworks à base de plugins pour les navigateurs (Flash, Flex, Silverlight, JavaFX, etc.)

Ce manque de solutions natives, conjointement à la multiplication des navigateurs Web, a grandement complexifié le développement d'applications Web. La réaction du W3C était donc très attendue. Après quelques années de préparation par un groupe de travail constitué par Apple, Mozilla et Opera, une version de travail de la spécification HTML 5 est sortie en janvier 2008. Ce groupe de travail, rejoint par Microsoft en août 2009, est censé apporter des réponses aux besoins non traités jusqu'ici, en matière d'applications Web.

## Principes de base de HTML 5

Jusqu'ici, les langages HTML et XHTML étaient distincts et définis par leur syntaxe. HTML 5 définit un langage unique. Ce langage :

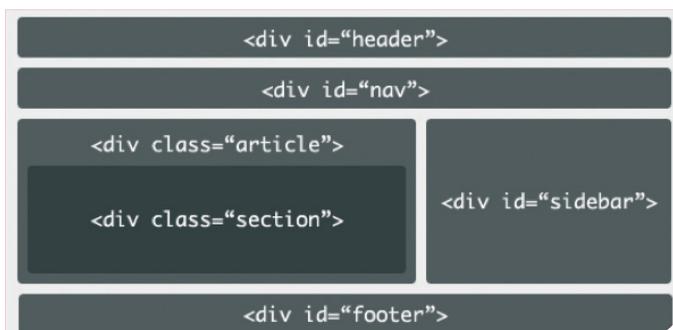
- est défini par sa structure DOM (Document Object Model, qui permet de représenter un document par une structure arborescente) et donc indépendamment de la syntaxe ;
- peut être écrit par une syntaxe HTML ou une syntaxe XML : il ne s'agit donc que de deux sérialisations possibles parmi d'autres.

Cette spécification définit de nouveaux éléments, attributs et API plus adaptés aux applications Web, ainsi que des consignes afin d'encourager des implémentations interopérables. Notez que HTML 5 est conçu pour être compatible avec ses anciennes versions : les navigateurs qui implémenteront la spécification HTML 5 devront toujours être capables d'interpréter les documents HTML plus anciens.

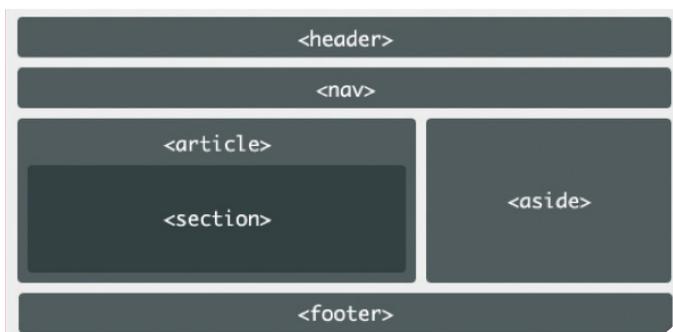
## Apports « basiques » de HTML 5

### Sémantique

Des éléments ont été ajoutés afin de donner plus de structure aux documents : section, article, aside, header, footer et nav, notamment. Leur nom est explicite quant à leur utilisation. Ainsi, on utilisera ces éléments natifs plutôt que de donner de la sémantique à des div (par exemple) en leur attribuant des identifiants ou classes CSS particuliers :



devient :



L'article <http://www.alistapart.com/articles/previewofhtml5> vous fait une excellente présentation de ces éléments. HTML 5 fournit également un élément *dialog* pour représenter une discussion, ainsi qu'un élément *figure* permettant d'ajouter un titre à un élément donné (une image, par exemple).

### Contrôles

Plusieurs nouveaux contrôles sont prévus.

#### - Champs de saisie

L'élément input est doté de nouvelles valeurs pour sa propriété type. Essentiellement : datetime, date, month, week, time, number, range, email, url, search, color.

Ainsi les navigateurs pourront proposer nativement l'interface utilisateur adéquate comme par exemple, un sélecteur de dates : [Fig.1] Les navigateurs prendront également en charge l'envoi au serveur non pas de la valeur saisie, mais d'une valeur formatée.

On retrouve ici un principe largement mis en œuvre dans la plupart des frameworks : si on reprend notre exemple du sélecteur de dates, l'utilisateur pourra choisir des dates qui lui seront présentées au format local (JJ/MM/AAAA pour la France) mais les données envoyées au serveur pourront être toujours de la forme AAAA/MM/JJ, après avoir été validées. Justement, l'attribut required permettra nativement de valider qu'un champ a bien été renseigné : [Fig.2]

### - Combo-boxes

La combo-box fait son apparition en HTML ! L'élément `datalist` permet en effet de bénéficier d'une combo-box native, et donc de s'affranchir des frameworks qui nous le fournissaient, sous le nom « auto-complete » ou « suggest ».

### - Datagrids

La datagrid fait également son apparition dans le langage HTML. Elle permet de représenter et manipuler des arbres, des listes et des tableaux. Une datagrid est initialisée à partir d'un élément HTML :

```
<datagrid>
  <table>
    <tr><td>1</td><td>Item 1</td></tr>
    <tr><td>2</td><td>Item 2</td></tr>
    <tr><td>3</td><td>Item 3</td></tr>
  </table>
</datagrid>
```

Dans cet exemple, la datagrid est peuplée à partir d'un tableau. Mais on aurait également pu utiliser un élément `select` afin de peupler la datagrid à partir d'une liste. Et même un ensemble d'éléments HTML, chacun représentant une ligne de la datagrid.

Il faut bien comprendre l'intérêt de la datagrid par rapport aux éléments HTML traditionnels : les éléments `table` et `select` ne sont pas manipulables par l'utilisateur, ils ne servent qu'à présenter des informations. La datagrid permet à l'utilisateur de les manipuler : sélectionner des lignes, des colonnes, des cellules, éditer des cellules, supprimer des lignes, des colonnes, trier les données, etc. Elle propose également une API qui permet de la manipuler programmiquement.

### - Menus

HTML définit un élément `menu` qui permet de définir des menus et des barres d'outils. Voici un exemple de barre d'outils :

```
<menu type="toolbar">
  <li>
    <menu label="File">
      <button type="button" onclick="fnew()">New...</button>
      <button type="button" onclick="fopen()">Open...</button>
      <button type="button" onclick="fsave()">Save</button>
      <button type="button" onclick="fsaveas()">Save as...</button>
    </menu>
  </li>
  <li>
    <menu label="Edit">
      <button type="button" onclick="ecopy()">Copy</button>
```

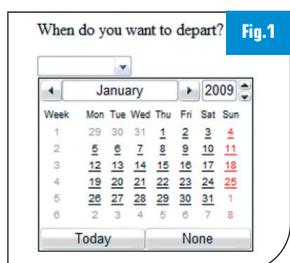


Fig.1



Fig.2

```
<button type="button" onclick="ecut()">Cut</button>
<button type="button" onclick="epaste()">Paste</button>
</menu>
</li>
</menu>
```

Pour créer un menu contextuel il suffirait de modifier l'attribut `type` du menu de `toolbar` à `context`. Les éléments peuvent alors référencer ce menu contextuel grâce à un nouvel attribut `contextmenu`.

Les éléments de ce menu peuvent être d'autres éléments menu (comme dans l'exemple ci-dessus), des séparateurs ou des commandes : une commande peut être représentée par un élément `a`, `button`, `input`, `option`, ou `command`.

Ce dernier élément `command` est apparu avec HTML 5. Il permet une sémantique plus fine :

```
<command onclick="undo()" label="Undo" />
```

### - Barres de progression

L'élément `progress` permet de présenter la progression d'une tâche :

```
Progress: <progress><span id="p">0</span>%</progress></p>
<script>
  var progressBar = document.getElementById('p');
  function updateProgress(newValue) {
    progressBar.textContent = newValue;
  }
</script>
```

Notez que l'appel à cette fonction `updateProgress()` devra être fait manuellement. Dans le même esprit, un élément `meter` est également prévu, pour afficher des mesures : pourcentages de votes, utilisation d'espace disque...

### Multimédia

HTML 5 fournit deux éléments `video` et `audio` dont l'objectif est de supporter nativement l'intégration et le contrôle des médias dans les pages Web :

```
<video src="video.mp4" id="video"></video>
<script>
  var video = document.getElementById("video");
</script>
<button type="button" onclick="video.play()">Play</button>
<button type="button" onclick="video.pause()">Pause</button>
<button type="button" onclick="video.currentTime = 0">Rewind</button>
```

Il sera donc possible de personnaliser l'interface utilisateur de manipulation des médias. Si ce n'est pas souhaité, l'attribut `controls` permettra d'indiquer au navigateur que c'est à lui de fournir cette interface. Ce sera aux navigateurs de choisir les bons codecs. Il sera possible de spécifier différents formats via l'élément `source` :

```
<video>
  <source src="video.ogv" type="video/ogv">
  <source src="video.mp4" type="video/mp4">
</video>
```

## gros plan \html

L'élément audio est prévu pour fonctionner d'une manière similaire :

```
<audio>
  <source src="audio.oga" type="audio/ogg">
  <source src="audio.mp3" type="audio/mpeg">
</audio>
```

Là encore, l'objectif est donc de fournir nativement ce que les solutions RIA nous ont proposé via des frameworks ou plug-ins.

### Compléments

Les frames disparaissent ! Les éléments *frame*, *frameset* et *noframes* ne sont pas dans la spécification HTML 5 car leur utilisation a rendu difficiles l'expérience utilisateur et l'accessibilité. Beaucoup d'attributs de présentation ont également été retirés afin d'inciter l'utilisation des équivalents CSS, bien plus adéquate : *align*, *bgcolor*, *border*, *cellpadding*, *cellspacing*, *width*, *height*...

## Apports « avancés » de HTML 5

### Drag and drop et copier/coller

Le « drag and drop », ou « glisser/déposer », est aujourd'hui un besoin courant dans les applications Web riches. Il est donc aujourd'hui largement proposé par la plupart des frameworks RIA. Il semblerait donc presque banal qu'il soit défini dans la spécification HTML 5. Cependant, il faut savoir que dans les navigateurs actuels, qui implémentent HTML 4, les opérations « drag » sont limitées : vous ne pouvez lâcher des éléments qu'à l'intérieur de la fenêtre du navigateur. Si cela n'est pas gênant dans beaucoup de cas, il s'agit là d'une limitation importante, qui vous empêche de partager des données d'une application Web à une autre d'une manière conviviale. HTML 5 a pour objectif de casser cette limitation en proposant des interfaces *DragEvent* et *DataTransfer* :

```
interface DragEvent : MouseEvent {
  readonly attribute DataTransfer dataTransfer;
}
```

De manière assez traditionnelle pour ce genre de problématiques, les événements suivants sont gérés au fur et à mesure du drag and drop : *dragstart*, *drag*, *dragenter*, *dragleave*, *dragover*, *drop*, *dragend*. Lors de l'événement *dragstart*, l'objet *DataTransfer* est initialisé à une valeur qui pourra être récupérée lors de l'événement *drop*. Cette valeur pourra contenir des informations concernant l'objet déplacé. Cet objet doit être marqué comme étant déplaçable (attribut *draggable*). Voici un exemple très simple permettant de déplacer un élément *div* d'un conteneur à un autre :

```
<div>
  <div id="drag"
    draggable="true"
    ondragstart="return dragStart(event)"
    ondragend="return dragEnd(event)">
    Item 'draggable'
  </div>
</div>

<div id="dropZone"
  ondrop="return dragDrop(event)"
```

```
  ondragover="return dragOver(event)">
</div>
```

On notera l'utilisation de nouveaux attributs dont le rôle est assez explicite. Voici le code des fonctions JavaScript qui traitent les événements générés :

```
<script>
function dragStart(event) {
  ev.dataTransfer.setData("ElemId", event.target.getAttribute('id'));
  return true;
}

function dragOver(event) {
  // Ici on peut aussi retourner 'true' si on souhaite empêcher
  // le dépôt sur l'élément survolé
  return false;
}

function dragEnd(event) {
  ev.dataTransfer.clearData("ElemId");
  return true;
}

function dragDrop(event) {
  var elemId = ev.dataTransfer.getData("ElemId");
  event.target.appendChild(document.getElementById(elemId));
  // On empêche la propagation de l'événement
  ev.stopPropagation();
  return false;
}
</script>
```

Petite particularité : la notion de drag and drop n'est pas définie explicitement dans la spécification. Cela ouvre la voie à de nombreuses possibilités, comme celle d'effectuer des « copier/coller ». Le site <http://html5demos.com/drag> propose quelques démonstrations : [Fig.3]

### Géolocalisation

Cet aspect n'est pas directement inclus dans la spécification mais il y est souvent lié. Une API de Géolocalisation est également en cours de spécifications : <http://www.w3.org/TR/geolocation-API/>. Son objectif est de permettre l'accès à la latitude et la longitude du périphérique sur lequel est exécuté le navigateur. Ces informations sont déterminées par analyse de diverses sources d'information : GPS, adresse IP, RFID, WiFi, Bluetooth, GSM... Voici un exemple d'utilisation de cette API :

```
function traiterPosition(position) {
  // Utilisation des coordonnées (ex : affichage d'une carte)
  // position.coords.latitude, position.coords.longitude
}

navigator.geolocation.getCurrentPosition(traiterPosition);
```

Cet exemple montre un appel unique à l'API : celle-ci propose un mécanisme de cache configurable en cas de besoin d'appels répétés. Là encore vous pourrez effectuer des tests vous-mêmes sur le site <http://html5demos.com/geo>. Les plus inquiets pourront constater

que le navigateur vous demande votre permission avant de récupérer vos coordonnées : [Fig.4]

## Graphisme 2D

HTML 5 définit un élément `canvas` qui permet de faire du dessin en JavaScript : formes, graphes, retouches photographiques, et même animations.

```
<canvas id="monCanvas" width="150" height="150"></canvas>
```

Les dessins se font en travaillant sur des *rendering contexts*. Nous resterons ici dans le cadre du contexte 2D, mais il existe également un contexte 3D (en cours de spécification, basé sur Open GL ES).

Voici le code permettant de dessiner un rectangle bleu :

```
<script>
  var canvas = document.getElementById("monCanvas");
  if (canvas.getContext) {
    var ctx = canvas.getContext("2d");

    ctx.fillStyle = "rgb(0,0,255)";
    ctx.fillRect (10, 10, 50, 50);
  }
</script>
```

Cet exemple est très simple mais cette API propose de nombreuses possibilités : rectangles, lignes droites, arcs de cercle, courbes cubiques et quadratiques, intégration d'images, translations, homothéties, rotations, gestion de la transparence, dégradés, ombres, animations, interactions avec la souris... [Fig.5]

Pour en savoir plus, vous pouvez par exemple consulter le site de Mozilla : [https://developer.mozilla.org/en/Canvas\\_tutorial](https://developer.mozilla.org/en/Canvas_tutorial).

## Gestion du Content-Type et de l'encodage des caractères

HTML 5 définit de nouveaux algorithmes de détection du Content-Type et de l'encodage des caractères. Du côté des développeurs, spécifier l'encodage utilisé devrait être bien plus simple. On retiendra les deux méthodes suivantes :

- Renseigner l'en-tête Content-Type des requêtes HTTP
- Renseigner un élément meta de la page avec l'attribut `charset` comme suit :

```
<meta charset="UTF-8">
```

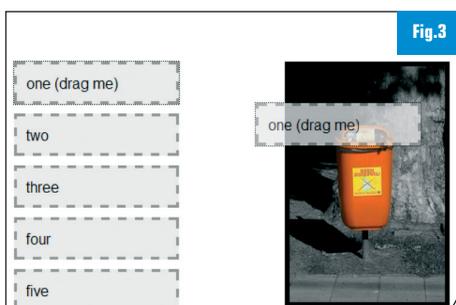


Fig.3

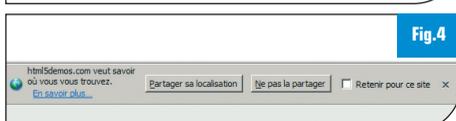


Fig.4



Fig.5

Plus besoin, donc, de la traditionnelle syntaxe verbeuse suivante :

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

Cette syntaxe sera cependant toujours supportée, toujours dans un souci de rétro-compatibilité.

## Gestion de l'historique et du bouton « Back »

HTML 5 définit la notion de *browsing contexts* : il s'agit d'environnements dans lesquels les objets Document sont présentés aux utilisateurs. Ainsi, une fenêtre possède un *browsing context*, tout comme un onglet, une *iframe* ou une *frame*... L'ensemble des documents d'un *browsing context* est nommé *session history*. Des objets *History* sont ainsi constitués, empilant les URL (ou des objets représentant un état) associées aux documents. Cet historique est manipulable via une API dédiée :

```
interface History {
  readonly attribute long length;
  void go(optional in long delta);
  void back();
  void forward();
  void pushState(in any data, in DOMString title, optional in DOMString url);
  void clearState();
};
```

Ainsi l'appel suivant suffit à revenir à l'état précédent de l'historique :

```
window.history.back();
```

Cette API ressemble fortement à celle qui existait auparavant. Elle devrait cependant être plus poussée grâce à cette notion d'état, qui facilitera la gestion de l'historique dans les applications AJAX, de la même manière que l'ont proposé les frameworks.

## Historique d'annulations

Il s'agit là d'une nouveauté qui n'est pas encore prête : la spécification nous indique que plusieurs points pourraient bien être à revoir, nous allons donc en rester à une présentation brève. L'idée de cette API nommée *UndoManager* est d'associer un historique d'annulations à chaque document. Cet historique sera manipulable et composé d'une liste d'éléments, pouvant être de deux types :

- *DOM changes* : il s'agit d'objets représentant des modifications de la structure DOM opérés suite à une manipulation utilisateur (changement du contenu d'un nœud ou de la valeur d'un de ses attributs) ;
- *Undo objects* : il s'agit d'objets représentant des informations de plus haut niveau (état avant une action utilisateur ou l'envoi d'une requête au serveur, par exemple).

## Mode offline (déconnecté)

HTML 5 est accompagné de la spécification *Offline Web Applications* : <http://www.w3.org/TR/offline-webapps/>.

Elle a pour objectif de permettre le développement d'applications Web utilisables en mode déconnecté, ou offline. Pour ce faire, elle apporte deux éléments :

- une base de données embarquée dans le navigateur et manipulable via une API reposant sur des requêtes SQL ;
- un cache applicatif permettant de stocker dans le navigateur les réponses des requêtes HTTP effectuées.

# gros plan \html

## Base de données

La base de données locale supporte les transactions. Elle sera manipulable de la manière suivante :

```
var db = openDatabase("notes", "", "The Example Notes App!", 1048576);

function renderNotes() {
  db.transaction(
    function(tx) {
      tx.executeSql(
        'CREATE TABLE IF NOT EXISTS Notes(title TEXT, body TEXT)',
        []
      );
      tx.executeSql('SELECT * FROM Notes', [], function(tx, rs) {
        for(var i = 0; i < rs.rows.length; i++) {
          // Traitement de la note : rs.rows[i]
        }
      });
    }
  );
}

function insertNote(title, text) {
  db.transaction(
    function(tx) {
      tx.executeSql(
        'INSERT INTO Notes VALUES(?, ?)', [ title, text ],
        function(tx, rs) {
          // ...
        },
        function(tx, error) {
          // Traitement de : 'sql', error.message
        }
      );
    }
  );
}
```

Cette API s'inspire très fortement de Google Gears. Sauf qu'au lieu d'un plugin, cette base de données sera fournie nativement par les navigateurs.

## Cache applicatif

Ce cache reposera sur l'attribut *manifest* de l'élément html, qui pointe vers l'URI d'un fichier nommé « manifest ».

Ce fichier indique simplement la liste des fichiers à mettre en cache (section « CACHE MANIFEST »), ainsi que ceux qui ne doivent pas être mis en cache (section « NETWORK ») :

```
CACHE MANIFEST
index.html
help.html
style/default.css
images/logo.png
images/background.png

NETWORK:
server.cgi
```

La spécification prévoit d'autres moyens de définir le cache, via un préfixe par exemple. Elle prévoit également une API permettant aux scripts d'ajouter et retirer dynamiquement des fichiers du cache.

## Aspects communs

Pour déterminer si le client est connecté ou non, il suffira d'invoquer :

```
var isOnline = navigator.onLine;
```

## Web workers

Cet aspect non plus n'est pas directement inclus dans la spécification HTML 5 mais souvent lié. Une API est également en cours de spécifications : <http://www.whatwg.org/specs/web-workers/current-work/>.

Cette spécification définit une API pour exécuter des scripts en tâche de fond, donc indépendamment des scripts de l'interface utilisateur. Elle permet ainsi l'exécution de scripts « longs » :

- sans interruption par des scripts de traitement d'actions utilisateur (clics par exemple) ;
- sans besoin de mécanisme permettant de conserver une page réactive.

Mais cette API est très claire : les Workers sont lourds et à utiliser avec parcimonie, et uniquement si réel besoin.

Voici un exemple très simple issu de la spécification. La tâche de fond est ici un algorithme naïf de recherche de nombres premiers :

```
The highest prime number discovered so far is:
<output id="result"></output>

<script>
  var worker = new Worker('worker.js');
  worker.onmessage = function (event) {
    document.getElementById('result').textContent = event.data;
  };
</script>
```

Voici le contenu du fichier worker.js :

```
var n = 1;
search: while (true) {
  n += 1;
  for (var i = 2; i <= Math.sqrt(n); i += 1)
    if (n % i == 0)
      continue search;
  // found a prime!
  postMessage(n);
}
```

La spécification regorge d'exemples variés, à complexité et intérêt croissants.

## Web sockets

HTML 5 introduit l'API Web Sockets, qui permettra de communiquer avec le serveur de manière bi-directionnelle à la manière de Comet et AJAX, mais avec les deux avantages suivants :

- cette API sera native ;
- elle ne nécessitera qu'une seule connexion avec le serveur, au lieu de deux dans le cas de Comet et AJAX.

En effet, Comet et AJAX reposent sur le protocole HTTP donc nécessitent une connexion pour envoyer des données et une autre

/update/2009/10

**ComponentSource**  
The Definitive Source of Software Components  
www.componentsource.com



## DXperience Enterprise à partir de € 896

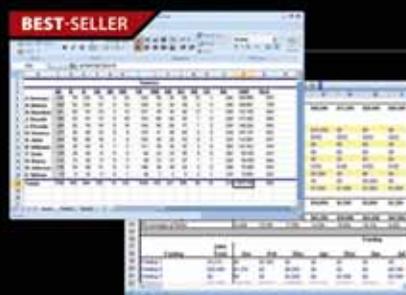
**DevExpress**

Une gamme complète de contrôles de présentation et de solutions de rapport pour vos nouvelles applications WinForms, ASP.NET et WPF.

Les composants incluent :

- Saisie de données – des grilles multidimensionnelles aux arborescences
- Analyse – des tables pivot OLAP aux jauges interactives et aux tableaux de bord
- Rapports – concepteur intégré pour les utilisateurs
- Graphiques – des douzaines de types de graphiques 2D et 3D
- Planification – contrôles de calendrier de type Outlook®
- Navigation – des rubans et des barres d'outils aux menus et aux docks
- Mise en page de formulaires – conception et personnalisation complète des formulaires
- Thèmes d'application – un ensemble complet d'enveloppes et de styles d'application
- Rendu des formulaires – impression WYSIWYG des contrôles DevExpress
- Filtrage des données – interfaces de génération de requêtes simples à complexes
- Contrôles d'utilitaires – de la vérification orthographique aux Assistants d'applications

Pour en savoir plus sur DXperience et la ligne de produits primés DevExpress, visitez :  
[www.componentsource.com/devexpress](http://www.componentsource.com/devexpress)



## FarPoint Spread for .NET Bundle à partir de € 1 134

**FarPoint**

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

- Nouvelle fonctions ASP.NET: extensions AJAX, impression vers PDF, éditeur de modèle de ligne, assistant de démarrage rapide, nouveaux types de cellules, etc.
- Nouvelle fonctions WinForms: export PDF, groupements, barre de formules, Excel 2007 XML
- Contrôle unique 2 milliards de feuilles avec 2 milliards de lignes et 2 milliards de colonnes
- Inclut une version pour .NET 2.0 et .NET 3.5 (Visual Studio 2008)



## ActiveReports 6 à partir de € 482

**GrapeCity**

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

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

© 1996-2009 ComponentSource. Tous droits réservés. Tous les prix sont corrects au moment de la presse. Prix en ligne mai différentes de celles décrites en raison de fluctuations quotidiennes et remises en ligne.

**Siège social en Europe**  
ComponentSource  
30 Greyfriars Road  
Reading  
Berkshire  
RG1 1PE  
Royaume-Uni

**Siège social aux États-Unis**  
ComponentSource  
650 Claremore Prof Way  
Suite 100  
Woodstock  
GA 30188-5188  
États-Unis

**Siège social au Japon**  
ComponentSource  
3F Kojimachi Square Bldg  
3-3 Kojimachi Chiyoda-ku  
Tokyo  
Japon  
102-0083

Numero vert:  
**0800 90 92 62**  
[www.componentsource.com](http://www.componentsource.com)

Nous acceptons les bons de commande. Contactez-nous pour demander un compte de crédit.





# DÉVELOPPEZ 10 FOIS PLUS VITE

PLATEFORME PROFESSIONNELLE  
DE DÉVELOPPEMENT (AGL)

Internet, Intranet, RAD, **WEB 2.0**, **PHP**  
Lien à toutes les bases de données

# WEBDEV

## CRÉEZ FACILEMENT DES SITES RELIÉS AUX BASES DE DONNÉES

**WEBDEV 14** est l'environnement de développement professionnel qui permet de développer **10 fois plus vite** tous les types de sites et d'applications **reliés aux données de votre entreprise**.

Le **WEB 2.0** est facile: l'activation d'**AJAX** dans vos sites s'effectue par un simple clic dans l'éditeur de code.

WEBDEV 14 est certainement le seul environnement au monde à proposer autant de souplesse et de puissance.



*Sous l'éditeur de code de WEBDEV 14: un clic et le traitement programmé devient «Ajax»*

**WEBDEV 14 gère le cycle complet de développement et d'administration** : WYSIWYG, langage LSG, générateur de code PHP, débogueur, Webservices, gestionnaire de sources, installateur, base de données SQL intégrée et lien avec toutes les bases du marché, composants, éditeur d'états PDF et code-barres, règles métier, dossier, outils de déploiement et d'administration...: tout est inclus, en français.

**Vous aussi, réalisez vos sites WEB 2.0 10 fois plus vite... avec WEBDEV 14.**

UN OCEAN DE WEB 2.0

**VOTRE CODE EST  
MULTI-PLATEFORMES:**

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



Compatible WINDEV



**DEMANDEZ LE DOSSIER GRATUIT**

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

Fournisseur Officiel de la Préparation Olympique

[www.pcsoft.fr](http://www.pcsoft.fr)



# Comment tester vos pages web ?

Les tests sont incontournables dans la vie quotidienne du développeur. Ils peuvent être de nature diverse et se situer à différents niveaux de l'application. Les plus simples, que tout développeur réalise, sont exécutés manuellement ou à l'aide d'un débogueur, après avoir codé une partie de l'application. Nous pouvons les définir comme des « tests d'intégration » manuels.

## Tests d'intégration manuels

Ces tests prennent une grande partie du temps de développement car le développeur est obligé d'exécuter l'interface graphique (vos pages web), d'effectuer une série d'actions sur cette interface (clic sur un bouton par exemple, passage d'une page à l'autre, enregistrement d'un formulaire, etc.) et de vérifier le résultat de cette action. Cela permet de tester les composants et des classes interagissant ensemble pour produire un résultat final. Cependant, à chaque modification du code nous sommes obligés de recommencer les tests car nous n'avons aucune garantie de n'avoir pas cassé une autre fonctionnalité ailleurs dans l'application. Si le test ne réussit pas, tous les composants et les classes sont en échec solidairement et il est parfois difficile de trouver la vraie source du problème. Le test d'intégration signifie que nous testons deux modules d'application ou plus comme un groupe. En outre, il n'est pas possible de les enregistrer et de les exécuter d'une manière automatique. Ceci nous amène à la réflexion concernant les « tests unitaires ».

## Tests unitaires

Depuis un certain temps dans l'industrie du software, la méthode TDD (Test Driven Development) s'impose. Pour résumer rapidement, il s'agit d'abord d'écrire le test unitaire, le faire échouer, d'écrire le code de production pour faire passer le test à tout prix, refactorer, vérifier que le test passe toujours, et ainsi de suite. Faire échouer le test nous assure que lorsqu'il passera, ce ne sera pas par hasard et que s'il y a un problème dans le code de production le test va échouer à chaque fois (si la partie fonctionnelle du code ne change pas, dans le cas contraire il faut également mettre à jour le code du test).

Un test unitaire est une partie de code (le plus souvent une méthode) qui fait appel à une autre partie de code et vérifie si certaines hypothèses préétablies sont correctes. Si ces hypothèses s'avèrent être fausses, le test unitaire a échoué. L'« Unité » est une méthode ou fonction. Il est également important de définir les caractéristiques d'un bon test unitaire :

- Doit être automatisé et répétable.
- Doit être facile à implémenter.
- Une fois écrit, il doit être réutilisable dans le futur.
- N'importe qui doit être capable de l'exécuter.
- On doit pouvoir l'exécuter sur un simple clic sur le bouton.
- Doit s'exécuter rapidement.

Les tests unitaires sont souvent utilisés pour tester la logique métier contenue dans les classes et les composants, la couche d'accès aux données, les services web, etc.

Ce qu'il faut retenir des tests unitaires, c'est que s'ils sont bien écrits et constamment maintenus et refactorés, ils améliorent grandement la qualité finale de la livraison. De plus, le risque de régres-

sion de code est limité car dès que quelque chose ne va pas, les tests unitaires nous le diront.

Mais qu'en est-il pour tester notre interface graphique ? Nous avons vu que les tests manuels de l'interface sont très longs et ne permettent pas de détecter toutes les anomalies. Les tests unitaires sont plutôt bons pour automatiser les tests de la logique métier. Mais heureusement, il existe des outils pour tester vos pages web d'une manière automatique. Nous parlons ici des « tests fonctionnels ».

## Tests fonctionnels

Les tests fonctionnels permettent de tester la logique contenue dans les pages. Les pages des sites web ont leur propre fonctionnement qu'il n'est pas facile de tester. Nous pouvons en citer quelques uns :

- Utilisation d'AJAX
- Frames (de moins en moins)
- Popups et les fenêtres de dialogue (confirm, login, etc.)
- Fenêtres de dialogue HTML (modales et non modales)

## Les outils pour tester les pages web

Les outils qui permettent de réaliser les tests fonctionnels ne sont pas nombreux mais les plus utilisés sont :

- Selenium
- WatiN
- Ivonna
- NUnitAsp
- Team System Web Test
- The Art Of Test

### Selenium

<http://selenium.openqa.org>

C'est un framework de tests afin d'automatiser les tests des applications web à travers des plates-formes multiples. Il a beaucoup de fonctionnalités et leur apprentissage peut être très long.

### WatiN

<http://watin.sourceforge.net>

C'est un framework inspiré de Watir (Web application testing in Ruby). Il est open source et permet d'écrire des scripts pour les actions des navigateurs (en C# par exemple).

### Ivonna

<http://sm-art.biz/ivonna.aspx>

Ivonna paraît très complet. C'est un framework de tests qui permet d'exécuter les tests ASP.NET sans utiliser la vraie session HTTP et les pages. La fonctionnalité également appréciable est la compilation des pages à la volée que vous souhaitez tester. Vous pouvez

donc tester les contrôles à l'intérieur de ces pages sans le navigateur. Le seul problème est qu'il faut disposer d'une licence de TypeMock Isolator Framework pour pouvoir en bénéficier (Ivonna a été écrit en collaboration avec Typemock).

### NUnitAsp

<http://nunitasp.sourceforge.net>

N'est plus maintenu.

### Team System Web Test

Il faut avouer que le Team System Web Test a des fonctionnalités intéressantes et paraît être un outil très puissant permettant par exemple d'enregistrer et de rejouer les requêtes web pour les pages et vérifier différentes choses pendant l'exécution. Il faut cependant disposer de Visual Studio Team Test ou Team Suite.

### TheArtOfTest

<http://www.artoftest.com>

C'est un framework de tests commercial dont la liste de fonctionnalités est très impressionnante comme par exemple l'intégration dans Visual Studio, support de Silverlight, support des composants tiers comme Telerik.

Afin d'illustrer tous ces propos, nous allons utiliser le framework WatiN pour créer notre premier test de pages web. Pourquoi WatiN ? Tout d'abord parce qu'il est open source, vous pouvez l'utiliser avec Internet Explorer ou Firefox, il est facile d'utilisation et permet de créer des tests automatisés pour les langages .NET.

## Utiliser WatiN et NUnit pour tester nos pages

WatiN est très facile d'utilisation, il permet d'ouvrir les instances d'un navigateur web et grâce à interop, il interagit avec les éléments dans la form web. En résumé, WatiN vous permet d'obtenir et de mettre à jour les valeurs des éléments des pages web, exécuter des événements, etc. La version utilisée dans cet article (WatiN 2.0 bêta 1 du 30/03/2009) supporte les navigateurs IE6, IE7, IE8 (il y a quand même quelque soucis mineurs), Firefox 2.x et 3.x, Chrome (expérimental).

### Prérequis

Avant de pouvoir réaliser notre premier test nous devons réaliser certaines étapes :

1. Téléchargez WatiN à partir du site : <http://sourceforge.net/projects/watin/files/>. La version la plus récente au moment de l'écriture de cet article est 2.0 bêta 1.
2. Nous aurons besoin de NUnit pour réaliser nos tests unitaires. Téléchargez une version sur le site <http://www.nunit.org/index.php?p=download>. La version la plus récente est NUnit 2.5.2.9222.
3. Si vous utilisez Internet Explorer en version inférieure à 8, l'outil Internet Explorer Developer Toolbar que vous pouvez télécharger sur le site de Microsoft sera très utile (<http://www.microsoft.com/downloads/details.aspx?FamilyID=e59c3964-672d-4511-bb3e-2d5e1db91038&displaylang=en>). Pour Firefox vous pouvez télécharger Firebug à l'adresse suivante <http://getfirebug.com/>. Ces outils vous permettront d'explorer les éléments de la page web afin de créer les tests d'automatisation plus facilement.
4. Visual Studio 2008 Express Edition

### Avant de commencer

Afin d'effectuer nos tests nous allons créer deux projets dans Visual Studio. Un site web ASP.NET où nous allons construire une petite page web que nous allons tester et un projet du type bibliothèque des classes où nous allons placer nos tests. Les deux projets peuvent se trouver dans la même solution mais cela n'est pas nécessaire. Nous pouvons nommer le site web « MyWebSite » et le projet des tests « MyWebSite.Tests »

Afin de pouvoir écrire des tests pour WatiN, il nous faut ajouter une référence dans le projet de tests vers la dll WatiN.Core.dll qui se trouve à l'endroit où vous avez installé WatiN après l'avoir téléchargé dans l'étape 1. Les espaces de noms disponibles sont :

- `WatiN.Core.Comparers` : Contient les différentes classes pour réaliser les comparaisons. Vous pouvez hériter de la classe `BaseComparer` pour implémenter votre propre comparateur.
- `WatiN.Core.Constraints` : Contient l'ensemble des classes qui servent à définir les contraintes lors de la recherche des éléments.
- `WatiN.Core.DialogHandlers` : Contient les classes qui permettent de gérer les boîtes de dialogue des navigateurs.
- `WatiN.Core.Exceptions` : contient les classes qui nous permettent de gérer les erreurs lorsqu'un problème se produit.
- `WatiN.Core.Interfaces` : contient les interfaces qui permettent d'étendre les fonctionnalités de WatiN comme par exemple `ILogWriter` qui vous permet d'implémenter votre propre logueur.
- `WatiN.Core.Logging` : contient les classes qui vous permettent de loguer les actions que votre code est en train de réaliser. Vous pouvez loguer par exemple dans le débogueur ou la console.

Il est également important d'ajouter une référence vers la dll `nunit.framework.dll` qui se trouve dans la GAC après l'installation de NUnit. Cela vous permettra d'écrire les tests et les faire exécuter dans l'interface graphique de NUnit. Nous avons donc ajouté les références suivantes : [Fig.1].

## Création de notre premier test d'une page web

### Création de la page web

Tout d'abord nous allons créer une page web très simple. Nous allons le faire dans notre projet « MyWebSite ». Admettons que c'est une page de login avec deux zones de texte et un bouton. Une zone de texte pour le login utilisateur, une pour le mot de passe et un bouton pour valider. Notre page peut ressembler à la page ci-dessous : [Fig.2]

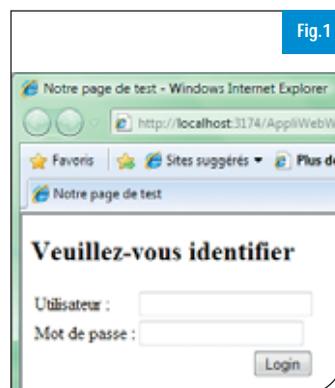


Fig.1

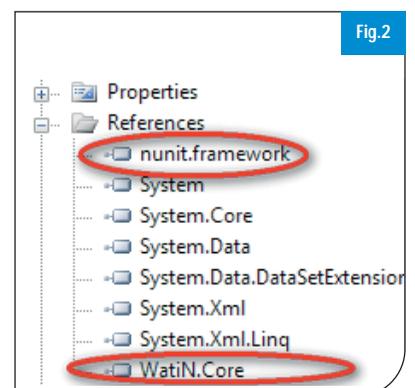


Fig.2

# webmaster \ outils

Le code de la page est très simple :

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default
.aspx.cs" Inherits="_Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>Notre page de test</title>
</head>
<body>
<form id="form1" runat="server">
<h2>Veuillez-vous identifier</h2>
<table>
<tr>
<td>Utilisateur :</td>
<td><asp:TextBox runat="server" ID="txtUtilisateur" /></td>
</tr>
<tr>
<td>Mot de passe :</td>
<td><asp:TextBox runat="server" ID="txtPassword" Text
Mode="Password" /></td>
</tr>
<tr>
<td colspan="2" align="right">
<asp:Button runat="server" ID="btnLogin" Text="Login"
OnClick="btnLogin_Click" />
</td>
</tr>
<tr>
<td colspan="2"><asp:Label runat="server" ID="lblResult"
/></td>
</tr>
</table>
</form>
</body>
</html>
```

Le codebehind de l'évènement click() du bouton Login est également très simple. Il vérifie si l'utilisateur et le mot de passe fournis en interface sont égaux respectivement à « admin » et « programmez » :

```
protected void btnLogin_Click(object sender, EventArgs e)
{
string utilisateur = txtUtilisateur.Text;
string mdp = txtPassword.Text;
```

```
// si l'utilisateur est "admin" avec le mot de passe
"programmez" alors
// un message de succès est affiché
if (utilisateur == "admin" && mdp == "programmez")
lblMessage.Text = "Vous êtes autorisé à entrer !";
else
lblMessage.Text = "Utilisateur et/ou le mot de passe ne sont
pas corrects";
}
```

## Ecriture de notre test

Nous allons écrire notre test dans le projet « MyWebSite.Tests ». Le premier test sera très simple. Nous allons vérifier que lorsqu'on passe un utilisateur et un login valide, lors du clic sur le bouton « Login » l'identification se passe correctement. Le code suivant le réalise :

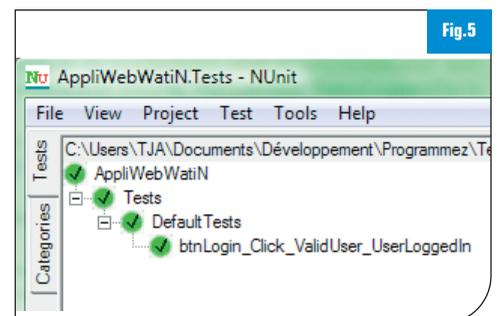
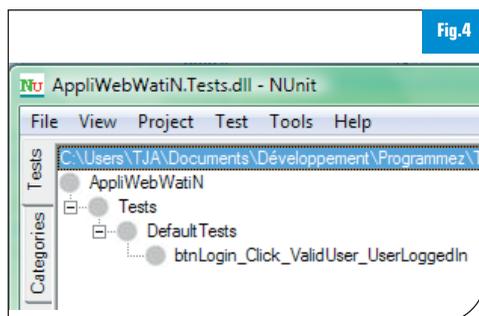
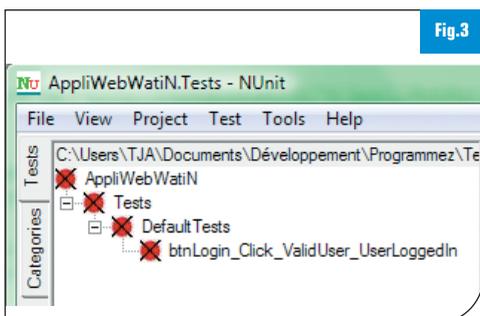
```
[TestFixture]
public class DefaultTests
{
[Test(Description = "Vérifier que l'utilisateur est logué")]
public void btnLogin_Click_ValidUser_UserLoggedIn()
{
// ouvrir la connexion vers la page web testée
using (IE ie = new IE("http://localhost:3174/AppliWebWatiN/"))
{
// renseigner la zone de texte utilisateur
ie.TextField(Find.ById("txtUtilisateur")).TypeText("admin");

// renseigner la zone du mot de passe
ie.TextField(Find.ById("txtPassword")).TypeText("admin");

// exécuter le click du bouton
ie.Button(Find.ById("btnLogin")).Click();

// vérifier la réponse, le test doit réussir
Assert.AreEqual(true, ie.ContainsText("Vous êtes autorisé
à entrer !"));
}
}
}
```

Maintenant, il suffit de compiler notre projet « MyWebSite.Tests » et de charger la dll de sortie du projet dans la console NUnit pour l'exécuter. Nous apercevons notre test CheckIfLoggedIn() dans la console NUnit : [Fig.3]



# DÉVELOPPEZ 10 FOIS PLUS VITE

PLATEFORME PROFESSIONNELLE  
DE DÉVELOPPEMENT (AGL)

Pocket PC, Smartphone, Terminal  
Windows CE, Windows Mobile

# WINDEV Mobile

## DÉVELOPPEZ VOS APPLICATIONS POUR POCKET PC, SMARTPHONE & TERMINAL MOBILE : FACILE !



**VERSION  
EXPRESS  
GRATUITE**  
Téléchargez-la !

Un tableau  
de bord  
temps réel sur  
votre portable?  
Facile !



**WINDEV Mobile 14** est l'environnement de développement professionnel qui permet de développer jusqu'à **10 fois plus vite** les applications sur mobile dont votre entreprise et vos clients ont besoin: gestion de stock, force commerciale, géolocalisation, saisies médicales, expertises, relevés de terrain, prise de commande temps réel, réglage de chaîne de production, etc...

La **puissance** et la **facilité** de développement de WINDEV Mobile 14 permettent un développement complet en quelques journées.

L'environnement est livré complet, le déploiement des applications réalisées est **gratuit** sans redevances (base de données incluse).

Toutes les fonctionnalités d'un AGL professionnel sont offertes. Tous les aspects de la mobi-

lité sont gérés: base de données, accès direct, réplication, WiFi, Bluetooth, 3G, Internet, socket, ActiveSync, réseau, J2EE, SMS, RFID, lien avec votre S.I., **codes-barres automatiques...**

***Vous aussi réalisez vos applications mobiles  
10 fois plus vite... avec WINDEV Mobile 14.***

(Logiciel **professionnel**, Support Technique gratuit\*)



Compatible WINDEV

**VOTRE CODE EST  
MULTI-PLATEFORMES:**

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

Logiciel **professionnel**

**DEMANDEZ LE DOSSIER GRATUIT**

252 pages + DVD + Version Express incluse  
+ 112 Témoignages.

Tél: **04.67.032.032** ou **01.48.01.48.88**  
info@pcsoft.fr

Fournisseur Officiel de la Préparation Olympique

[www.pcsoft.fr](http://www.pcsoft.fr)



## webmaster \ outils

Avant d'exécuter le test, assurez-vous que votre page web est disponible à l'adresse que vous passez au constructeur de la classe IE dans le test :

```
IE ie = new IE("http://localhost:3174/AppliWebWatiN/");
```

Maintenant il suffit de l'exécuter en cliquant sur le bouton « Run » pour vérifier si notre test passe : [Fig.4]

Le test n'a pas réussi alors qu'il aurait dû.

Le message d'erreur nous informe que lorsqu'on utilise WatiN, le Thread.Apartmentstate qui exécute le test doit être en mode STA. Par défaut NUnit exécute ses tests en multithread (MTA), il faut donc lui indiquer de les exécuter en STA. Pour cela, nous pouvons ajouter un fichier de configuration à notre projet de tests pour indiquer à NUnit d'exécuter les tests en mode STA. Voici le fichier de configuration :

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
    <sectionGroup name="NUnit">
      <section name="TestRunner" type="System.Configuration.
NameValueSectionHandler"/>
    </sectionGroup>
  </configSections>
  <NUnit>
    <TestRunner>
      <!-- Valid values are STA,MTA. Others ignored. -->
      <add key="ApartmentState" value="STA" />
    </TestRunner>
  </NUnit>
</configuration>
```

Maintenant, si nous recompilerons et réexécutons les tests tout devrait se passer comme prévu : [Fig.5].

A titre d'information, il est également possible de démarrer le Web Server à partir du code, ce qui nous évite de l'ouvrir avec Visual Studio. Cela permet l'exécution des tests indépendamment de l'interface graphique.

### Les boîtes de dialogue

WatiN permet également de tester les actions sur les boîtes de dialogue des navigateurs. WatiN 2.0 bêta 1 a introduit la nouvelle classe DialogHandlerHelper qui permet de déterminer quel objet spécifique doit être utilisé avec quelle boîte de dialogue. Dans l'exemple ci-dessous, nous créons une boîte de dialogue de type « confirm » et nous vérifions que le bouton OK a bien été cliqué :

```
[Test(Description = "Vérifier qu'une boîte de dialogue s'affiche
et que le bouton OK est cliqué")]
public void btnAlert_Click_ConfirmDialog_ShouldClickOK()
{
  // ouvrir la connexion vers la page web testée
  using (IE ie = new IE("http://localhost:3174/AppliWebWatiN/"))
  {
    ie.ShowWindow(WatiN.Core.Native.Windows.NativeMethods.WindowS
```

```
howStyle.Maximize);

  // notre handler pour la boîte de dialogue
  var confirmDialogHandler = new ConfirmDialogHandler();

  // UseDialogOnce indique que nous allons
  // manipuler une fois notre boîte de dialogue
  using (new UseDialogOnce(ie.DialogWatcher, confirmDialog
Handler))
  {

    // exécuter le click du bouton
    ie.Button("btnAlert").ClickNoWait();

    // attendre que la boîte de dialogue existe
    confirmDialogHandler.WaitUntilExists();

    // et nous cliquons sur le bouton OK
    confirmDialogHandler.OKButton.Click();

  }

  // nous attendons que IE termine le traitement
  ie.WaitForComplete();

  Assert.AreEqual(ie.DialogWatcher.Count, 0, "Il ne doit y
avoir de boîtes de dialogue ouvertes");
}
```

### Support des expressions lambda

Avec l'arrivée de C# 3.0 les expressions lambda sont devenues très populaires. Nous pouvons également les utiliser comme l'alternative à la fonction Find de WatiN. Nous pouvons par exemple remplacer le code suivant :

```
ie.TextField(Find.ById("txtPassword")).TypeText("programmez");
```

par l'équivalent en expression lambda :

```
ie.TextField(txt => txt.Id == "txtPassword").TypeText("programmez");
```

### Pour aller plus loin

Il existe un outil pour faciliter la création des tests unitaires. Il s'agit de WatiN Test Recorder que vous pouvez télécharger à l'adresse : <http://watintestrecorder.sourceforge.net/>. Il vous permettra d'enregistrer les actions que vous effectuez dans le navigateur et ensuite de générer le test unitaire que vous pouvez ajouter dans votre projet de tests. Bien que l'outil présente encore quelques bugs (tout du moins avec IE8), il s'avère cependant très facile d'utilisation et très puissant. WatiN permet également de tester les sites avec de l'AJAX. Cela s'avère indispensable sachant que la plupart des sites web utilisent cette technologie.

### Références

WatiN: <http://watin.sourceforge.net/>

WatiN Test Recorder: <http://watintestrecorder.sourceforge.net/>

■ Thomas Jaskula

Chef de projet .NET - <http://blogs.developpeur.org/tja>

# Les recrutements sont en baisse

## Comment les développeurs gèrent-ils la crise ?

Budgets informatiques revus à la baisse, afflux sur le marché d'informaticiens expérimentés et de jeunes diplômés. Pour tirer parti du temps libre – chômage ou inter-contrats – la formation est en première ligne.

**A**près un rallongement des durées d'inter-contrats, de nombreuses SSII ont décidé de baisser leurs effectifs. Dans le secteur « ingénierie informatique », les intentions en matière d'effectifs sont négatives, d'après l'étude de conjoncture de la Banque de France (juillet 2009). Tendence confirmée par l'Apec qui met en évidence une chute de 38%, sur les sept premiers mois de l'année, du nombre d'offres d'emploi déposées concernant le secteur informatique. Depuis novembre 2008, l'Apec enregistre en effet un net recul dans la diffusion des offres. Pour les 1er et 2e trimestres 2009, le Baromètre Manpower faisait état d'un solde net d'emploi, corrigé des variations saisonnières, négatif. Stable par rapport au 1er trimestre 2009, il marque une baisse de 5 points comparé au 2e trimestre 2008. Il s'agit du solde net le plus faible enregistré par la France depuis le lancement du Baromètre dans ce pays au 3e trimestre 2003.

### La situation de l'emploi

En informatique comme dans beaucoup d'autres secteurs, les indicateurs d'emploi sont au rouge. Pour la première fois en 4 ans, le nombre des demandeurs d'emploi en informatique, qui diminuait chaque mois, est reparti à la hausse depuis juillet 2008. De plus, selon l'organisme statistique du ministère du Travail, le nombre de demandeurs d'emploi inscrits à Pôle emploi de catégorie A (catégorie à laquelle se rattachent les informaticiens) a augmenté de 25,6% sur un an, faisant passer à la profession la barre symbolique des

5% de chômeurs. L'informatique de gestion est le segment professionnel le plus touché par la crise avec une baisse de 44% des offres sur la même période. Derrière ce segment, les autres fonctions informatiques sont à peine plus protégées avec une chute de 34 % du nombre d'offres dans l'informatique industrielle et de 40% pour les fonctions exploitation/maintenance et systèmes/réseaux. Symboliquement, le Munci (Mouvement pour une union nationale et collégiale des informaticiens) s'était fixé une cote d'alerte à 25 000 informaticiens au chômage. Elle est désormais dépassée.

Pourtant, l'informatique se porte, aujourd'hui encore, mieux que d'autres domaines. Lesjeudis.com relève que « *le marché de l'emploi des ingénieurs a commencé à s'essouffler en janvier 2009, toutefois des niches subsistent notamment dans le secteur de l'informatique.* » On observe, en effet, un décalage par rapport aux autres secteurs. Ce qu'explique **Nicolas Sadirac**, directeur de l'Epitech : « *Les instruments de mesure de l'emploi sont décalés dans le temps. En septembre 2008 on ne voyait pas encore les répercussions de la crise sur l'informatique ; cela n'a commencé qu'en mars-avril 2009. Par ailleurs, les cycles de projets sont d'environ 5 ans.* » **Alexandre Xiradakis**, directeur marketing et communication du groupe Les Jeudis, observait en juin un ralentissement important du nombre d'offres : de -30 à -40%. Avec toutefois un espoir d'éclaircie : « *On assiste sur la fin du mois d'août à une reprise de la croissance du nombre d'offres sur lesjeudis.com,*



la variation étant de -10% par rapport à août 2008 », précise-t-il.

Des SSII comme le groupe Hélice, continuent à recruter. « *Nous sommes constamment à la recherche de nouveaux profils pour rester réactifs et anticipatifs des besoins éventuels de nos clients, avec cette même volonté de partager avec nos futures recrues nos talents et notre progression* », affirme **Catherine Maistret**, directeur général du groupe Hélice. Les profils qu'elle recherche : administrateurs et ingénieurs système Unix, Linux, Windows, réseaux ; analystes et ingénieurs d'exploitation Unix, Mainframe, Windows ; administrateurs de bases de données Oracle ; ingénieurs d'études et analystes Java/JEE, Microsoft, Mainframe, client-serveur, décisionnel ; chefs de projets et directeurs de projets études et production. « *Le recrutement concerne aussi bien des jeunes diplômés que des expérimentés issus de formation Bac+2 à Bac+5.* »

Du fait de la crise, le recrutement est plus facile pour ces sociétés, la difficulté est donc du côté des candidats : comment améliorer son « employabilité » ?

### Inter-contrats et chômage partiel

La situation économique se traduit par des processus de recrutement allongés, passant de un à plus de trois mois en un semestre, selon lesjeudis.com, ainsi que par des inter-contrats plus nombreux et plus longs. Selon Régis Granarolo, président du

Remise des diplômés promo 2008 (Supinfo).



Nicolas Sadirac, Epitech

## carrière \\\ emploi

Munci, « en plus de l'éclatement de la bulle des offres d'emploi et du gel des recrutements, les taux d'intercontrats ont quasiment doublé en moyenne. »

Pour affronter la crise qui touche certains de ses clients, notamment dans le secteur automobile, la société de conseil et d'ingénierie Altran a mis en œuvre des mesures de chômage partiel. Pour **Emmanuel Arnould**, président du Geicet (Groupement européen d'ingénierie et de conseil en technologies) « Le chômage partiel est une pratique très inhabituelle dans le secteur. Il est une réponse à la crise. Une manière de garder ses collaborateurs en ces temps difficiles et d'éviter les mesures particulièrement violentes comme les licenciements ».

Les développeurs indépendants sont évidemment concernés directement par la situation. En effet, les deux tiers des freelance ayant répondu à l'enquête lancée par la CyberGazette (le journal en ligne des freelance) reconnaissent avoir subi un effet de la crise, dont la conséquence moyenne (pour ces 2/3) est une perte de 30 % du chiffre d'affaires. Les manifestations de la crise le plus souvent citées sont : « négociation du prix d'une mission plus difficile que d'habitude », et « difficultés inhabituelles à trouver de nouvelles missions ». D'autres réponses sont plus inquiétantes : « fin prématurée d'une mission en cours », « obligation de rabais sur une mission en cours », « impossibilité de trouver de nouvelles missions »...

### Quelles perspectives ?

Si la tendance se confirmait, cela devrait avoir pour conséquence une explosion du nombre d'informaticiens au chômage pour cet automne. Une fois épuisées les périodes d'intercontrats et autres tentatives de gestion d'effectifs, les SSII pourraient procéder à des dégraissages plus ou moins massifs. Par ailleurs, si la confiance dans une reprise d'activité remonte un peu, elle devrait se faire dans un contexte de prix revus à la baisse. **Régis Granarolo** (Munci) observe « des mouvements étranges en interne, comme la multiplication

des missions déqualifiantes ». Cependant, une plus faible activité n'est pas nécessairement un inconvénient pour tous. « La baisse d'activité peut arranger certaines SSII qui étaient en situation de surcharge, ce qui leur posait un vrai problème de recrutement jusqu'en 2008. D'où le recours à l'offshore, par nécessité », analyse Nicolas Sadirac (Epitech). Par ailleurs, Alexandre Xiradakis observe que « les entreprises commencent à entrevoir la reprise de la demande pour début 2010. Pour ne pas rater cette opportunité, elles doivent amorcer au plus vite le recrutement des candidats 'passifs' qui sont les profils les plus durs à recruter. »

### Comment s'en sortir ?

« Les profils techno s'en sortent plutôt bien : spécialistes en libre ou Microsoft .Net ; SAP résiste bien, Java moins », analyse Alexandre Xiradakis. Les entreprises recherchent surtout des personnes expérimentées, les débutants ont moins de chance. Le secret est la polyvalence. On demande aux développeurs une double compétence : technique et métier, ou bien plusieurs techniques. Les DSI évoquent l'amélioration de la communication entre informatique et métiers (91%), la création de postes et de processus d'interface avec les métiers (72%), le développement de la double compétence métier et technique des collaborateurs SI (69%). Alexandre Xiradakis insiste aussi sur l'ouverture nécessaire pour pouvoir réorienter le profil plus facilement. Ce que confirme **Stéphane Potdevin**, directeur d'agence du groupe Hélice : « La personnalité est très importante, avec le sens du service, l'écoute, la rigueur et l'adaptabilité. Auparavant, nous acceptions de prendre des profils qui ne correspondaient pas à 100% à nos besoins. Aujourd'hui, nous recrutons sur les mêmes critères, mais nous éliminons ceux qui ne correspondent pas à 100%. » Côté compétences techniques, le marché des experts pointus n'est pas saturé, selon Nicolas Sadirac (Epitech). En effet, même si des profils experts comme les développeurs C++, architectes ou experts des tech-

nologies Java et JEE ne sont plus préservés de la crise, d'après lesjeudis.com, ils restent très convoités par des sociétés en plein essor, positionnées sur des segments en développement.

### Les jeunes diplômés fragilisés

Les difficultés touchent surtout les jeunes diplômés, tous secteurs confondus. Pour Régis Granarolo (Munci), la forte croissance du nombre de chômeurs en juillet est une demi-surprise. « Nous observons traditionnellement une hausse du nombre de chômeurs en été dans le secteur. Là, visiblement, ce sont les jeunes diplômés qui trinquent et on assiste également depuis quelques semaines à pas mal de ruptures conventionnelles dans les entreprises. » Habituellement très alertes au moment de chercher à séduire les jeunes diplômés, les SSII sont cet été particulièrement discrètes sur ce front, laissant présager des débuts difficiles pour les promos 2009.

L'informatique est moins concernée que les autres secteurs, même si, comme le précise Alexandre Xiradakis, les entreprises recherchent moins des débutants que des personnes expérimentées. Et côté embauche, les jeunes diplômés se retrouvent en concurrence avec ceux qui sont sortis des SSII. Une enquête menée par B2EN et lesjeudis.com met en évidence un « recentrage vers des profils déjà expérimentés (2 à 7 ans d'expérience), les profils junior ou non-informaticiens ont plus de mal à se positionner dorénavant. » Les entreprises sont devenues plus sélectives : « Auparavant, elles formaient les personnes qu'elles embauchaient », rappelle Alexandre Xiradakis.

Toutefois, les entreprises qui recrutent sont souvent à l'affût de compétences pointues dans les technologies actuelles. Ainsi, parmi les ingénieurs et consultants recrutés par la société Alti, la moitié concerne les jeunes diplômés. « Nous recherchons des ingénieurs spécialisés dans les technologies Java, Microsoft, ainsi que des consultants SAP », souligne Michel Hamou, directeur général du



Stéphane Potdevin,  
Hélice

# egilia®

## LEARNING

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

Faire de vos succès  
notre réussite

# www.egilia.com

CONTACTEZ NOS CONSEILLERS FORMATION

 **N° National 0 800 800 900**

APPEL GRATUIT DEPUIS UN POSTE FIXE

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



## carrière \ emploi



Marius Fodoreanu,  
Supinfo

groupe. « *Nous recherchons des jeunes diplômés issus d'écoles d'ingénieurs, Miage, mais aussi des jeunes diplômés issus d'écoles de commerce et universités pour les postes de consultants* ».

### La clé du succès : diplôme et ouverture d'esprit

Les écoles spécialisées en informatique ont toujours la cote : les entreprises recherchent des ingénieurs issus de telle ou telle école. De sorte que les élèves de ces écoles (Epita, Epitech, Supinfo, par exemple) res-

tent relativement optimistes. Même les demandes de stages (sauf éventuellement celui de la première année) trouvent facilement preneur, avec des stages de fin d'études débouchant couramment sur un CDI. « *Tous les étudiants de la promo 2008 ont trouvé un CDI à la sortie, même si les salaires sont souvent inférieurs* », précise **Marius Fodoreanu**, sorti de Supinfo l'année dernière. « *Depuis 2003, j'ai travaillé successivement au sein du groupe Bel, de l'OCDE (où je suis en ce moment), à l'AMF et chez Deloitte Conseil* ». Il

semble que la promo 2009 de Supinfo ne rencontre pas plus de difficultés que celle de 2008. « *J'ai intégré une équipe R&D d'une entreprise spécialisée en télé-médecine, à la suite d'un projet de l'école* », indique **Xavier Martelette**, qui passe ainsi directement de l'école au monde de l'entreprise, sans aucun temps mort. **Chris Roualin**, lui, a obtenu un CDI chez un prestataire de service avant même d'être sorti de l'école ; à sa sortie, son statut à l'embauche, au niveau bac+4, sera automatiquement requalifié en ingénieur.

## Se former, en attendant la reprise

Faute d'emploi, la solution qui s'impose est la formation. Pour ne pas perdre du temps, et surtout améliorer son « employabilité ». Selon une enquête de Kelformation, réalisée en partenariat avec Cadremploi, 80% des cadres considèrent leur formation professionnelle comme le meilleur rempart contre le chômage. Plus de la moitié d'entre eux envisageraient même de financer eux-mêmes leur formation. « *Que les cadres soient prêts à financer leur formation est une tendance avérée depuis plusieurs années en ce qui concerne les formations diplô-*

« *La formation professionnelle est aujourd'hui l'objet de toutes les attentions. Pour répondre aux difficultés du marché de l'emploi, les politiques mettent en place des mesures permettant à l'individu d'être acteur de son parcours professionnel. La nouvelle réforme va renforcer le Droit Individuel à la Formation et pousser les salariés à se former tout au long de leur vie professionnelle.* »

Les demandes de DIF concernent la bureautique, la formation technique et le développement personnel. Par

stages mettant l'accent sur le développement personnel, tels que « manager par gros temps », « accepter le changement », « maîtriser le stress »..., censés aider à répondre aux fortes pressions, objectifs chan-

des entreprises : « *Dans le cadre du contexte économique actuel, on constate une très forte croissance des demandes de formation dans le cadre du DIF (droit individuel à la formation) et notamment suite aux*

« *La société en crise libère du temps disponible pour les employés, qu'ils peuvent utiliser pour se former à de nouvelles connaissances.* »

Xavier Brouder, consultant formation chez Sogeti

« *La société en crise libère du temps disponible pour les employés, qu'ils peuvent utiliser pour se former à de nouvelles connaissances.* », confirme **Bruno Zefferi**, Directeur Lamy Formation & Liaisons Sociales Formation. Mais on ne peut pas l'affirmer pour les formations courtes. En revanche, jobetic.net note que « *les demandes de renseignements ne proviennent plus uniquement des services RH mais directement de salariés* ». Selon **Arnaud Gastaud**, responsable marketing de Kelfor-

exemple, les SSII forment leurs collaborateurs en inter-contrat, à des technologies telles que les ERP (SAP ou Microsoft) ou le développement Web (.net ou java). **Hubert Barkate** observe une recrudescence de demandes pour le « re-skilling », avec des cursus certifiants. Les organismes de formation se sont adaptés aux demandes des entreprises avec des

« *Les profils techno s'en sortent plutôt bien : spécialistes en libre ou Microsoft .Net ; SAP résiste bien, Java moins* »

Alexandre Xiradakis, Les Jeudis.com

geants, difficultés d'appréhension de l'environnement.

« *La société en crise libère du temps disponible pour les employés, qu'ils peuvent utiliser pour se former à de nouvelles connaissances* », confirme **Xavier Brouder**, consultant formation



chez Sogeti. Dans les SSII, l'inter-contrat est un moment propice pour se former :

« *Il faut être prêt pour la reprise ; nous formons des professionnels sur nos offres afin d'anticiper le démarrage* », renchérit **Serge Gatefait**, directeur du département formation chez Sogeti. Hubert Barkate, président de l'organisme de formation Adhara, note l'impact de la crise sur le comportement formation

« *licenciements économiques.* »

Pour répondre à cette demande de formation en inter-contrats, la SSII Ogmyos a lancé des packages de formation IT anticrise. « *Le contenu des formations existait déjà, mais nous avons aménagé notre catalogue avec des formations plus courtes, des dates de sessions plus fréquentes, sur deux sites* », indique **Damien Particelli**, directeur général d'Ogmyos. « *Ces nouveaux cursus permettront aux entreprises de capitaliser sur des phases de moindre activité et de mettre à jour les connaissances des collaborateurs sur les dernières innovations technologiques dans différents domaines. Elles permettront également de tirer parti des périodes inter-contrats des salariés* »,

« Nous n'avons pas eu d'alerte particulière pour la promotion 2008 », confirme **Thierry Boidard**, directeur général de Supinfo. « La plupart des étudiants étaient déjà en CDD ou CDI dès avril 2008 avec des salaires moyens d'embauche cohérents avec les précédentes promotions. » Ce que confirme Nicolas Sadirac, directeur de l'Epitech : tous les étudiants sortis en septembre 2008 ont trouvé du travail, même si le salaire moyen à l'embauche a baissé d'environ 1%.

### Revoir à la baisse ses prétentions salariales ...

« Les besoins en compétences sont toujours là. Les SSII ont ralenti leurs recrutements, mais les entreprises utilisatrices offrent toujours des emplois informatiques », poursuit Thierry Boidard. A la sortie, les étudiants ont couramment 2, 3 ou 4 propositions d'embauches, même si les entreprises sont moins enclines à se bousculer au portillon. A l'Epitech, on constate que le recrutement d'in-

génieurs informaticiens est particulièrement dynamique en région.

Les prétentions salariales doivent être revues à la baisse : « Les jeunes diplômés arrivant sur le marché du travail en 2009 ne doivent pas prendre pour référence les salaires obtenus par la promotion précédente car le contexte de marché est complètement différent », avertit **Robert Half**. « Ils sont devenus un peu plus modestes en matière de salaires », confirme Stéphane Potdevin (Hélice), qui recommande aux débutants de ne pas être trop présomptueux et ne pas se spécialiser dès la sortie d'école, alors qu'ils ne connaissent pas encore les métiers. « Tous les ingénieurs ont à peu près les mêmes compétences. Ce que nous demandons à un candidat, c'est l'ouverture d'esprit à tout ce que nous allons lui proposer : Soyez ouverts à tous les postes en tant que débutant ! », conseille Stéphane Potdevin. Par ailleurs, plutôt que de rechercher un CDI à tout prix, **Olivier Gélis**, directeur général de Robert Half Inter-

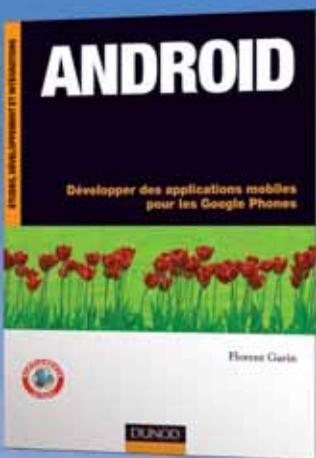
national France, conseille de privilégier la poursuite des études, des stages longue durée à l'étranger, voire l'expatriation. Une autre option est d'accepter des missions en intérim plutôt qu'un poste fixe, ce qui permet de « coller » à l'évolution du marché. « L'intérim est un véritable tremplin vers l'embauche en CDI », estime-t-il. S'il est encore trop tôt pour se prononcer en ce qui concerne les promotions 2009, les étudiants de Supinfo constatent d'ores et déjà qu'il n'y a pas de difficulté particulière pour trouver des stages en France. A l'étranger, au contraire, cela s'est avéré beaucoup plus problématique, car la culture de stage est beaucoup moins développée dans les pays anglo-saxons qu'en France.

■ Claire Rémy

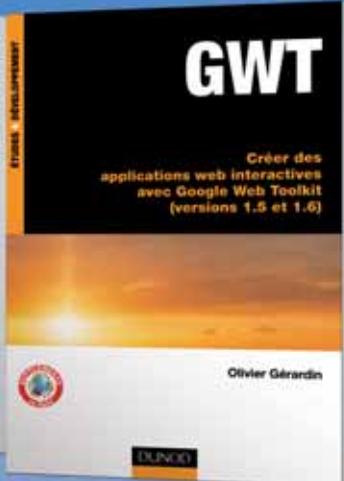
#### ERRATUM

Une erreur s'est glissée dans l'article « la formation, un domaine en plein essor » du n° précédent (Programmez 122). La photo figurant en p.45 est celle de Xavier BROUDER, consultant formation chez Sogeti (cf. encadré Formateur : un métier prisé).

## Maîtrisez les technologies informatiques



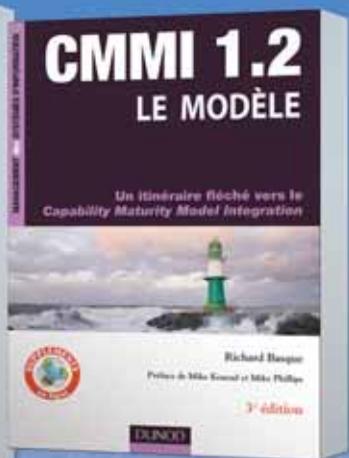
9782100531813 • 256 pages • 29 €



9782100531820 • 224 pages • 25 €  
À paraître en novembre



9782100530823 • 240 pages • 25 €



9782100522330 • 272 pages • 37 €  
À paraître en novembre

Tous nos ouvrages sont disponibles en librairie

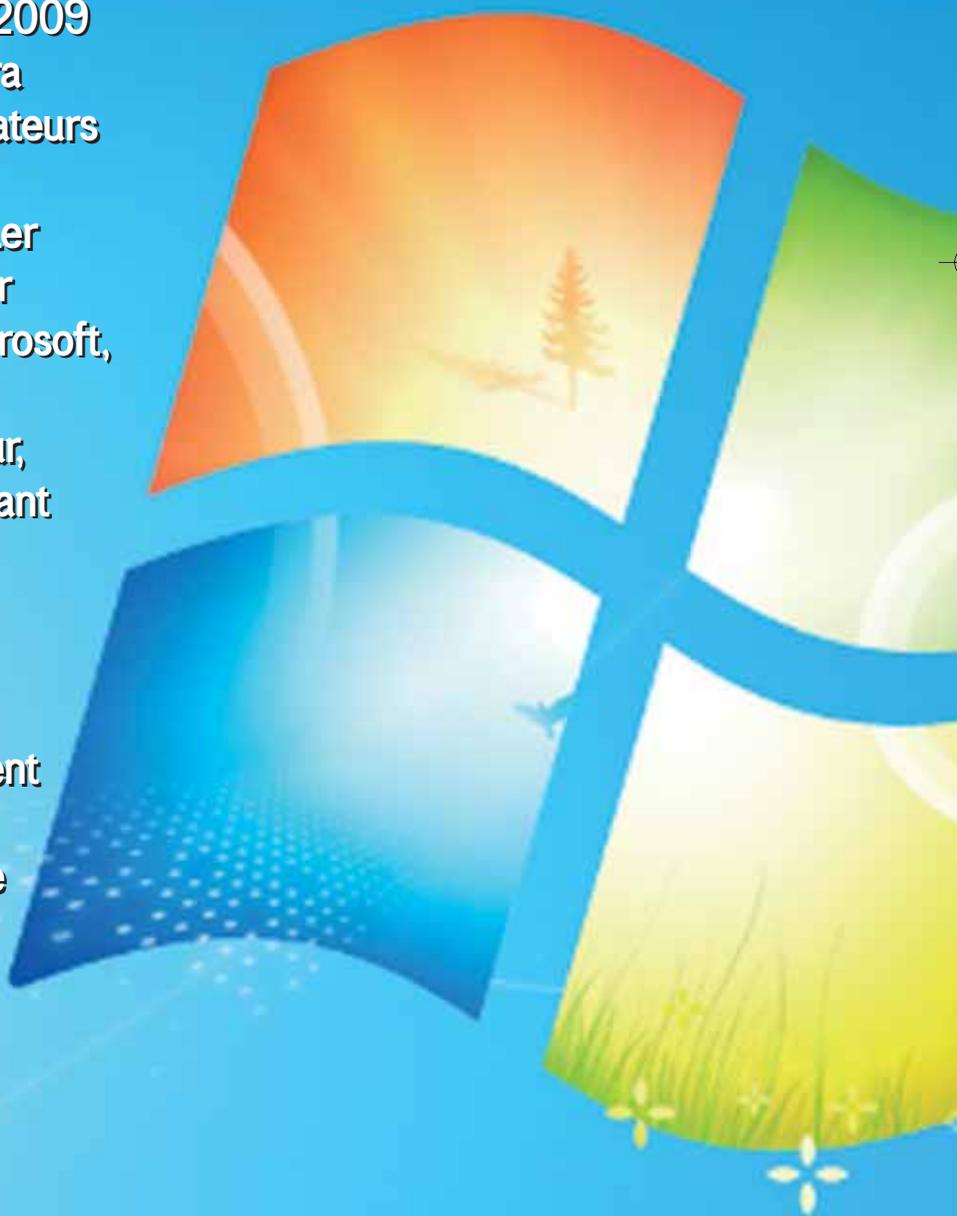
Réalisation : MATEO

Catalogue complet  [dunod.com](http://dunod.com)

  
**DUNOD**  
ÉDITEUR DE SAVOIRS

# Windows 7 : le guide du développeur

L'une des attentes de l'année 2009 touche à sa fin. Windows 7 sera disponible pour tous les utilisateurs le 22 octobre prochain et le développeur peut déjà l'installer et l'utiliser depuis août dernier (MSDN, Technet). Si pour Microsoft, l'enjeu est crucial, il l'est tout autant pour le développeur, car il doit migrer le code existant ou du moins vérifier son bon fonctionnement. Si le poste de développement migre en Windows 7, il faudra s'assurer du bon comportement des outils de développement, des SDK. Changer de système n'est jamais anodin.





# pour 1re partie

Officiellement, les commentaires sont plutôt optimistes sur la transition, la compatibilité.

Windows 7 modifie tout de même certains appels système, certaines fonctions impactant directement le code :

- API CoStartOutlookExpress est désactivé, du fait du retrait du système de Outlook. Microsoft recommande de retirer tous les appels d'interface à Windows Mail. Idem pour Windows Movie Maker
- Utilisation de GetNLSVersion Ex ou GetNLSVersion à la place de NLSVersion
- Modification sur le User Agent String dans Internet Explorer. Nécessité de bien préciser MSIE 8.0.
- Pour les versions 64-bit, recompilation des pilotes, des applications.
- Etc.

Les changements peuvent apparaître marginaux mais cela peut faire crasher une application. Et au-delà des modifications d'appels ou des API, il est nécessaire de respecter les nouveaux guides d'interface, les nouvelles fonctions.

Car Windows 7 introduit de nombreuses modifications que le développeur doit vérifier et le cas échéant intégrer dans son code, l'exemple le plus emblématique concerne la nouvelle barre des tâches comme nous l'avons vu dans le numéro précédent. Pour tous ces éléments, lisez impérativement : Windows 7 and Windows Server 2008 R2 Application Quality Cookbook (<http://code.msdn.microsoft.com/Windows7AppQuality>). Autre document incontournable, hormis notre grand dos-

sier, le Windows 7 Developer Guide (<http://code.msdn.microsoft.com/Win7DeveloperGuide>).

## Vérifier outil par outil

Avant de se lancer dans l'aventure Windows 7, il faut tester le bon comportement de ses outils de développement, de production puis envisager de migrer son poste de développement. Les principaux outils posent peu ou pas de problèmes : WAMP, MySQL (hormis quelques soucis en 64-bit cet été), SpringSource Tool Suite, RAD Studio 2010 (CodeGear), Eclipse. Netbeans pose quelques problèmes d'installation et d'instabilité en version RTM. Si vous n'êtes pas certain d'une compatibilité, attendez la mise à jour officielle de l'outil.

Vous pouvez vous rendre sur <http://www.microsoft.com/france/windows/windows7/compatibilite/> pour voir si l'outil est indiqué dans l'annuaire et son niveau de compatibilité. Le cas échéant, contactez l'éditeur, la communauté.

Dans cette première partie, nous allons aborder des notions fondamentales pour comprendre Windows 7 : le nouvel API Pack, les bibliothèques multitouch et de biométrie, le rôle des services déclenchés. Dans la 2e partie, nous aborderons les nouvelles API de déploiement et d'installation, DirectX, les nouveautés C++, PowerShell 2, le 64-bit. Dans les prochains mois, nous reviendrons naturellement régulièrement sur Windows 7.

Bonne découverte !

■ François Tonic

# La compatibilité des applications

Il n'existerait pas de problème de compatibilité si les nouvelles plates-formes, matérielles ou systèmes d'exploitation, ne changeaient pas. Or elles sont condamnées à évoluer pour apporter des améliorations. Dans le cas de Windows 7 par exemple, le nouveau système est plus sécurisé, plus robuste et plus ergonomique pour une meilleure expérience de l'utilisateur. Ces changements sont basés sur des améliorations introduites dans Vista.

Celui qui migre son poste depuis Vista bénéficiera de quelques améliorations appréciables, et celui qui migre depuis XP encore plus, ça, c'est la bonne nouvelle ! La mauvaise, c'est que ces changements peuvent parfois introduire des problèmes de compatibilité, et les applications qui fonctionnaient auparavant peuvent cesser de le faire, en totalité ou en partie sur le nouveau système.

Qu'on se rassure, un des objectifs majeurs de l'équipe de développement de Windows est de faire en sorte que la compatibilité applicative soit forte entre Windows Vista et Windows 7. Et on constate en effet, que peu d'applications déjà compatibles avec Vista sont incompatibles avec Windows 7. La situation est par contre assez différente pour les applications qui n'ont pas été adaptées depuis Windows XP...



Logo de certification "Compatible with Windows 7"

## Bien se préparer

La démarche de mise en compatibilité des applications vers Windows 7 est relativement simple sur le plan conceptuel. Il s'agit d'une démarche itérative :

- Installez votre application sur Windows 7
- Déroulez l'ensemble des tests (unitaires, automatisés, manuels, fonctionnels, ... tout ce que vous avez sous la main)
- Constatez les dégâts. Dans le meilleur des cas, tout fonctionne parfaitement, et vous avez terminé.
- Sinon il vous faut investiguer la cause des problèmes identifiés au cas par cas, corriger votre code, et réitérer.

Il existe des outils permettant d'accélérer l'identification des problèmes, et de gérer les cas où vous êtes dans l'impossibilité d'intervenir au niveau du code source de votre application (sources perdues, produit tiers, etc..) : on abordera ce point ultérieurement. Microsoft France organise régulièrement des **ateliers de compatibilité**. Ces ateliers d'un jour sont gratuits, et dédiés aux éditeurs de logiciels. Vous testez la compatibilité de vos applications en étant accompagné d'un expert technique, qui vous guidera à travers la démarche. Ces ateliers peuvent accueillir des professionnels non-éditeurs selon certaines conditions. Visitez le site <http://www.microsoft.com/fr-fr/isv> pour en savoir plus.



Cycle de mise en compatibilité d'une application avec Windows 7

Certains développeurs préféreront déboguer leurs applications directement sur Windows 7. Se pose alors la question de la compatibilité des outils de développement. Sachez que l'ensemble des outils qui ont été adaptés pour Windows Vista ont également été testés sur Windows 7, et ils fonctionnent convenablement. Vous subirez néanmoins les mêmes contraintes que sur Vista, comme lancer l'outil de développement en mode Administrateur pour certains scénarios.

Dans le monde Microsoft, cela concerne Visual Studio 2008 SP1, ainsi que Visual Studio 2005 SP1. J'ai également eu l'occasion d'adapter, sans grande difficulté et directement sur Windows 7, des applications développées dans des environnements plus « anciens », tels que Visual C++ 6.0, Visual Basic 6.0 et 5.0. Windows 7 livre en standard le Runtime .NET Framework 3.5 SP1. En attendant la sortie de Visual Studio 2010, l'environnement de développement idéal est donc le suivant :

- Visual Studio 2008 Service Pack 1
- Microsoft Windows SDK for Windows 7 and .NET Framework 3.5 SP1
- Windows API Code Pack for Microsoft .NET Framework

## Principales sources d'incompatibilité

D'une façon générale, les changements impactant la compatibilité des applications peuvent être regroupés en 3 catégories différentes. Certains changements sont liés aux évolutions « naturelles » du système d'exploitation. Il s'agit par exemple de modifications du numéro de version ou de structures internes du système. Certains autres changements sont imposés par les besoins d'innovation. Dans ce cas l'éditeur, en l'occurrence Microsoft, doit équilibrer les inconvénients liés au fait que certaines applications vont cesser de fonctionner, avec les bénéfices apportés par les améliorations en termes de sécurité par exemple, robustesse ou ergonomie. Finalement, certains autres changements sont introduits par la volonté de se conformer à des règles et standards externes, comme dans le cas d'Internet Explorer pour ne citer qu'un exemple. Cela peut malheureusement aller au détriment de la compatibilité des applications existantes.



L'ensemble des changements sont répertoriés dans le document intitulé « Windows 7 and Windows Server 2008 R2 Application Quality Cookbook ».

### Contrôle de comptes utilisateur (UAC)

L'élément le plus impactant sur la compatibilité s'appelle User Account Control (UAC ou Contrôle de comptes utilisateur en français). Si vous connaissez Windows Vista ou Windows 7, vous avez probablement déjà vu ces boîtes de dialogue qui s'affichent à chaque fois qu'un utilisateur ou qu'une application essaie d'exécuter une action potentiellement dangereuse, ou qui pourrait impacter une partie critique du système.

Depuis Vista, Windows introduit une notion de sécurité selon le principe du « moindre privilège ». L'idée consiste à combattre les virus et chevaux de Troie. Mais c'est également une manière de protéger les applications contre des changements de configuration par d'autres programmes. Il s'agit d'un mécanisme visant à protéger l'utilisateur et conserver le système dans un état connu et stable.

Ainsi, les API qui pourraient impacter la sécurité du système échouent (et retournent l'erreur `ERROR_ACCESS_DENIED`), que l'utilisateur connecté soit administrateur ou non, car ce dernier est considéré comme un « utilisateur standard » avec des privilèges réduits. Afin d'exécuter ces actions sans échec, il est nécessaire de procéder à ce que l'on appelle une « élévation de privilèges ». Cette opération peut être effectuée de plusieurs façons comme on le verra plus loin, et se manifeste visuellement par ces fameuses boîtes de dialogue qui demandent l'approbation ou le consentement de l'utilisateur connecté.

Le problème est que les anciennes applications ont été écrites avec la supposition que l'utilisateur est administrateur. Très souvent ces applications font des appels d'API qui exigent un niveau de privilèges plus élevé que nécessaire. C'est pourquoi lorsqu'elles sont exécutées en tant qu'utilisateur standard (le mode par défaut), ces applications échouent sous Vista ou Windows 7.

### Impact de UAC dans les anciennes applications

Les échecs les plus courants avec UAC :

- Ecriture dans la racine du disque, Program Files ou system32.
- Ecriture dans HKLM\Software
- Création d'objets kernel dans l'espace de nom global
- API Win32 impactant la configuration du poste

### Virtualisation UAC

Afin de simplifier la transition depuis XP, Vista et Windows 7 proposent une technique de mitigation basée sur une redirection des appels. On parle également de « virtualisation » des fichiers ou base de registre. Concrètement, lorsqu'une application 32-bit écrit dans un emplacement protégé par UAC, l'appel est redirigé vers une ressource dans l'espace privé de l'utilisateur appelant.

Les écritures dans les emplacements suivants :

- HKLM\Software
- %SystemDrive%\Program Files
- %WinDir%\System32
- %SystemDrive%

Sont redirigées vers :

- HKCU\Software\Classes\VirtualStore
- %LocalAppData%\VirtualStore

### Pratiques de développement

Avec UAC les bonnes pratiques de développement relèvent en réalité du bon sens, et d'une sensibilité à la sécurité. Voici quelques règles qu'il convient de respecter :

- Sauvegarder les préférences de l'utilisateur dans un espace qui lui est dédié, partagé pour tous les utilisateurs, ou réservé à votre application
  - Données utilisateur local : %LOCALAPPDATA%
  - Données utilisateur itinérant : %APPDATA%
  - Données applicatives : %ALLUSERSPROFILE%
  - Données partagées : %PUBLIC%
- Effectuer la configuration de l'application directement lors de l'installation (le programme d'installation est exécuté avec les privilèges élevés), plutôt qu'à la première exécution
- Ne pas sauvegarder de données volatiles directement dans le répertoire Program Files ou la base de registre HKLM
- Séparer les tâches administratives de l'application dans un processus séparé (qui sera exécuté séparément et avec un niveau de privilèges élevé).

Le dernier point ci-dessus est important. Car si la majorité des tâches dans une application classique ne requièrent pas de privilèges en particulier, certaines actions doivent être exécutées en mode élevé.

Dans ce cas, la recommandation consiste à recompiler ces actions dans un processus séparé que l'on lancera avec une élévation de privilèges. Il conviendra également de prévenir l'utilisateur par une signalétique graphique (l'icône du bouclier) que cette action va déclencher une demande d'approbation.

Voici un exemple .NET pour lancer un processus avec élévation de privilèges :

```
static void Main(string[] args)
{
    System.Diagnostics.Process process = new System.Diagnostics.
    Process();
    process.StartInfo.FileName = "c:\\windows\\system32\\notepad.exe";
    process.StartInfo.Verb = "runas";
    process.Start();
}
```

Et son équivalent Win32 :

```
#include <windows.h>

int _tmain(int argc, _TCHAR* argv[])
{
    LPTSTR szFilename = _T("c:\\windows\\system32\\notepad.exe");
    HINSTANCE hMod = ShellExecute(GetDesktopWindow(), _T("runas"),
    szFilename, NULL, NULL, SW_SHOWNORMAL);

    return 0;
}
```

### Manifest UAC

D'une façon générale, il est conseillé d'indiquer au système le niveau de privilèges désiré par l'application. Cela permet à Windows de lancer l'application avec le bon niveau de sécurité, et surtout évite les erreurs liées à UAC et les échecs de sécurité

# dossier \ windows 7

associés. Ainsi, si vous savez que votre application requiert une élévation de privilège (typiquement le cas d'une application d'administration système, ou un module applicatif dédié au paramétrage par exemple), vous pouvez indiquer cette information dans un fichier de manifest tel que dans l'exemple ci-dessous :

```
<?xml version="1.0" encoding="utf-8"?>
<asmv1:assembly manifestVersion="1.0" xmlns="urn:schemas-microsoft-com:asm.v1" xmlns:asmv1="urn:schemas-microsoft-com:asm.v1" xmlns:asmv2="urn:schemas-microsoft-com:asm.v2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
  <assemblyIdentity version="1.0.0.0" name="MyApplication.app"/>
  <trustInfo xmlns="urn:schemas-microsoft-com:asm.v2">
    <security>
      <requestedPrivileges xmlns="urn:schemas-microsoft-com:asm.v3">
        <!-- UAC Manifest Options -->
        <requestedExecutionLevel level="asInvoker" uiAccess="false" />
        <requestedExecutionLevel level="requireAdministrator" uiAccess="false" />
        <requestedExecutionLevel level="highestAvailable" uiAccess="false" />
        <!-- -->
        <requestedExecutionLevel level="requireAdministrator" uiAccess="false" />
      </requestedPrivileges>
    </security>
  </trustInfo>
</asmv1:assembly>
```

Ce manifest peut être incorporé à l'application de différentes façons :

1. En déployant un fichier votre\_appli.exe.manifest avec votre application (si votre application n'incorpore pas déjà un manifest intégré).
2. En créant un nouvel élément appelé « app.manifest » directement dans votre projet .NET avec Visual Studio 2008. Ce fichier sera fusionné au binaire lors de la compilation automatiquement.
3. En incluant ce fichier dans les ressources (\*.rc) de votre application :

```
#define MANIFEST_RESOURCE_ID 1
MANIFEST_RESOURCE_ID RT_MANIFEST "app.manifest"
```

## Changement de numéro de version interne de Windows

Le numéro de version interne de Windows 7 est 6.1 (par opposition à 5.1 pour XP et 6.0 pour Vista).

Les API *GetVersion* (natif Win32) et *System.Environment.OSVersion* (.Net) retournent désormais ce numéro lors de leur appel depuis Windows 7.

Ce changement anodin est important car les données télématiques recueillies pendant les phases Bêta et RC de Windows 7 montrent que de nombreuses applications refusent de se lancer après un test de version. C'est particulièrement vrai pour les utilitaires et les logiciels ludiques.

Voici un exemple d'appel pour une application .Net :

```
static void Main(string[] args)
{
    System.Version version = System.Environment.OSVersion.Version;
    Console.WriteLine("Version : {0}.{1}", version.Major, version.Minor);
}
```

Et Win32 :

```
#include <windows.h>
#include <tchar.h>

int _tmain(int argc, _TCHAR* argv[])
{
    OSVERSIONINFO version = {0};
    version.dwOSVersionInfoSize = sizeof(version);

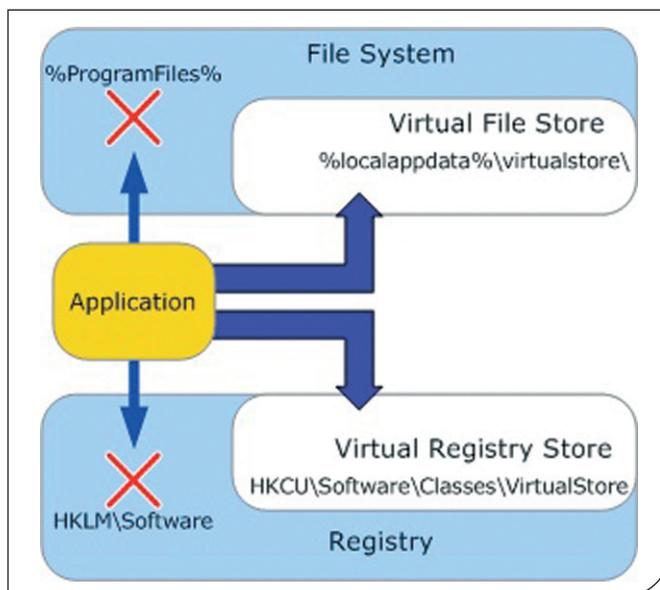
    if (FALSE != GetVersionEx(&version))
    {
        _tprintf(_T("Version : %d.%d\n"), version.dwMajorVersion, version.dwMinorVersion);
    }

    return 0;
}
```

## 64-bit

Les nouvelles machines que l'on trouve dans le commerce sont équipées de processeurs 64-bit en grande majorité. C'est évident dans le monde de l'entreprise mais ce changement devient également une réalité chez le particulier. Et Windows 7 est également disponible en version 64-bit. Mais est-ce que vos applications sont prêtes pour le 64-bit ?

Rassurez-vous. Windows 7 64-bit propose un environnement d'exécution 32-bit (WoW64), permettant aux applications 32-bit de fonctionner de manière transparente (hors pilotes de périphé-



Virtualisation UAC en action



riques). Ceux d'entre vous qui n'ont pas encore entamé la migration vers le 64-bit ont encore un peu de répit.

Attention toutefois aux applications bâties sur un Framework capable de cibler les 2 environnements, tel que le Framework .NET. Windows ne sait pas mixer les binaires 32-bit et 64-bit au sein du même processus.

Ainsi, si votre application est compilée pour s'exécuter dans les 2 environnements (configuration AnyCPU), vous pouvez rencontrer des surprises. En effet, votre processus .NET peut être exécuté en 64-bit, et le chargement d'une DLL 32-bit échouera. Ce scénario est courant avec des composants tierces parties (bibliothèques graphiques, pilotes de base de données, etc...).

Ce problème peut se résoudre facilement de différentes manières :

- Vous contactez l'éditeur de vos composants, afin d'obtenir une version 64-bit à déployer.
- Vous recompiliez votre exécutable en forçant une exécution en 32-bit (vous pouvez également utiliser l'utilitaire GFLAGS.EXE s'il s'agit d'une application .NET).

### Suppression de Windows Mail

Outlook Express ou Windows Mail étaient livrés dans les versions précédentes de Windows... On aime ou on n'aime pas, mais c'était bien pratique et certains éditeurs de solutions ont développé des dépendances à ce type d'outils pour la gestion des contacts ou de la messagerie électronique. Windows 7 ne livre plus ce type d'application en standard.

Si vous êtes dans ce scénario, vous devez inviter vos clients à basculer sur l'outil de leur choix. Le client de messagerie Windows Live Mail est un bon choix : il est compatible MAPI et gratuit. Il fait partie de la suite Windows Live Essentials.

### Quelques outils bien utiles

Microsoft distribue depuis quelques années des outils d'assistance à la migration pour aider les éditeurs de logiciels et les entreprises. Application Compatibility Toolkit (ACT) est une collection d'outils dédiés à la migration de parcs matériels et logiciels. ACT dépend d'un autre outil appelé Application Verifier (AppVerifier). Ces outils sont téléchargeables gratuitement sur le site Web de Microsoft. Les versions minimales requises pour Windows 7 sont les suivantes :

- Application Compatibility Toolkit 5.5
- Application Verifier 4.0



Boîte de dialogue de consentement UAC (application non-signée)

ACT propose plusieurs outils, plus ou moins utiles en fonction de votre contexte. Les grandes entreprises peuvent réellement bénéficier de ces outils qui aident à structurer le référencement des machines et des applications à migrer. Mais les développeurs peuvent également tirer parti de certains modules très spécialisés comme Standard User Analyser.

- **Application Compatibility Manager** : gestionnaire de configuration et référencement des logiciels, pilotes, périphériques et machines à migrer. Particulièrement utilisé par les DSI des grandes entreprises pour structurer et préparer la migration du grand parc informatique.
- **Compatibility Administrator** : Outil de gestion des Shim Fixes (voir plus bas)
- **Internet Explorer Compatibility Test Tool** : Active le mode « logging » dans Internet Explorer 8, et remonte toutes les erreurs de compatibilité liées à l'affichage des pages Web et la navigation.
- **Setup Analysis Tool** : Analyse dynamique des programmes d'installation basés sur Windows Installer ou produits tiers.
- **Standard User Analyser Wizard et Standard User Analyser** : Outils d'aide à la détection des problèmes liés à UAC. Permet également de concevoir des bibliothèques de Shim Fixes.

Mais que sont ces Shim Fixes, utilisés par ACT ? Il s'agit de petits morceaux de code spécialisés dans l'interception d'appels systèmes, dont le but consiste à en changer le comportement. Par exemple, le Shim Fix WinXPSP2VersionLie intercepte les appels à l'API GetVersionEx, et fait croire à l'application qu'elle s'exécute sur un système Windows XP SP2.

Il existe un nombre important de Shim Fixes dans le système, configurables de multiples façons et préconfigurés pour de nombreuses applications.

Microsoft livre régulièrement de nouveaux Shim Fixes ou des intercepteurs préconfigurés pour certaines applications de manière transparente via Windows Update.

Vous avez probablement utilisé des Shim Fixes sans le savoir. Lorsque vous choisissez un mode de compatibilité dans la fenêtre de propriété pour un raccourci d'application, vous utilisez en réalité une combinaison de Shim Fixes existants.

Vous pouvez créer de nouvelles bibliothèques de Shim Fixes pour corriger le comportement de vos applications, et déployer ces bibliothèques depuis Windows Installer. Il est toujours préférable de corriger le problème à la source, mais lorsqu'on ne possède pas le code source des produits, ces intercepteurs sont bien utiles.

### Et ensuite ?

Une fois que vos applications sont compatibles avec Windows 7, faites le savoir ! Si vous êtes éditeur de logiciels, vous serez probablement intéressé par le nouveau programme de certification « Compatible with Windows 7 ». Ce programme a été revu et simplifié : la certification devient désormais accessible au plus grand nombre. Faites certifier vos applications et profitez des bénéfices partenaires. Vous pouvez également faire connaître vos produits en renseignant le statut de compatibilité de vos applications sur le site web prévu à cet effet : <http://www.compatible-windows7.com>.

■ Stéphane Crozatier (stephanc@microsoft.com)

Partner Technology Advisor chez Microsoft France. Il travaille quotidiennement avec les éditeurs de logiciels pour les accompagner dans leurs choix et décisions technologiques.

# Découvrez le Windows API Code Pack

Windows 7 sera disponible officiellement dès le 22 Octobre prochain, et comme à chaque sortie d'une nouvelle version de Windows se posent les mêmes questions : faut-il mettre à profit les nouveautés du système, et comment le faire simplement ?

L'objectif du *Windows API Code Pack* est de faciliter ce choix :

- Il vous permet, lorsque votre application s'exécute sur un système Windows 7, d'utiliser les nouveautés.
- Il en simplifie l'utilisation grâce à une interface .Net, minimisant la nécessité de recourir au pinvoke.

Alors, pourquoi attendre ?

## Windows API Code Pack

Disponible sur le site MSDN, le *Windows API Code Pack* (WACP) est un ensemble de classes permettant d'accéder à certaines fonctionnalités de Windows Vista et/ou 7 qui ne sont à ce jour pas encore directement accessibles depuis le Framework .NET classique, notamment :

- La barre des tâches 7 (Jump Lists, Previews, Progress Icon)
- Les librairies de l'explorateur (7)
- Les task dialogs (Vista, 7)
- L'application recovery (Vista, 7)
- Le support des sensors comme la lumière ou les accéléromètres (7)
- Le support étendu des langues, de DirectX, les informations de consommation électrique, des configurations réseau...

Nous allons couvrir dans cet article des sujets chers aux applications de type « client riche » : le support de la nouvelle barre des tâches, des librairies, des TaskDialogs et l'application recovery.

Le WACP est composé principalement de deux assemblies : `Microsoft.WindowsAPICodePack` et `Microsoft.WindowsAPICodePack.Shell`. Chacune d'elle est majoritairement constituée de classes wrapper .NET autour des API systèmes de Windows, permettant aux développeurs .NET d'accéder à ces fonctionnalités système en utilisant des idiomes familiers.

Prenons par exemple le cas de la TaskDialog, dont l'utilisation n'est par défaut prévue qu'en C++ :

```
HRESULT TaskDialog(
    HWND hWndParent,
    HINSTANCE hInstance,
    PCWSTR pszWindowTitle,
    PCWSTR pszMainInstruction,
    PCWSTR pszContent,
    TASKDIALOG_COMMON_BUTTON_FLAGS dwCommonButtons,
    PCWSTR pszIcon,
    int *pnButton
);
```

Le WACP expose cette API avec notamment la méthode suivante, bien plus parlante pour un développeur .NET :

```
public static TaskDialogResult Show(string text, string instruction
Text, string caption)
```

La première des assemblies, `Microsoft.WindowsAPICodePack` (dénotee *Core*) forme la base de WACP. Elle expose les méthodes natives principales ainsi que les types de base nécessaires pour l'accès aux fonctionnalités système visées par le code pack. Les fonctionnalités mises à disposition par cette assembly sont : l'Application Restart, les TaskDialogs, les Network Lists, et la gestion d'énergie.

La deuxième assembly, `Microsoft.WindowsAPICodePack.Shell`, dépend de la première et met à disposition les fonctionnalités propres au shell, à savoir :

- La barre des tâches 7
- Les boîtes de dialogue Ouvrir/Sauvegarder/Sauvegarder sous...
- Les boutons CommandLink (avec icône, titre, descriptif, et support du « bouclier » UAC)
- Les dossiers favoris (KnownFolders)
- L'accès aux icônes système

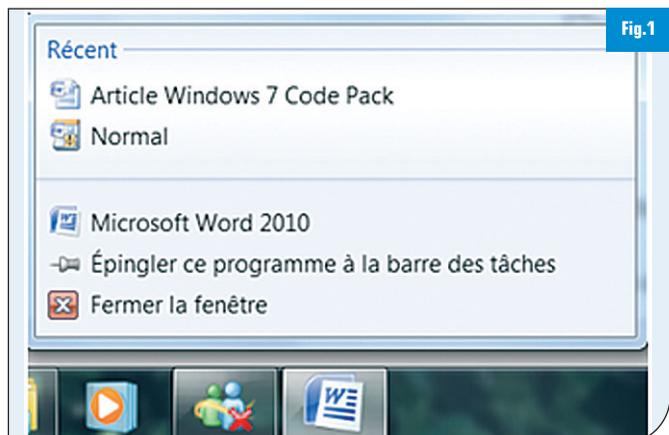
La solution WACP dispose également de deux autres assemblies `Microsoft.WindowsAPICodePack.ExtendedLinguisticServices` et `Microsoft.WindowsAPICodePack.Sensors`, dédiées respectivement au support étendu des langues et aux capteurs.

## La barre des tâches de Windows 7

De toutes les nouvelles expériences utilisateur que propose Windows 7, la nouvelle barre des tâches est sans doute celle qui sera la plus remarquée des utilisateurs. Outre un accès plus pratique aux applications, la barre des tâches de Windows 7 permet des interactions jusqu'alors inédites aux applications sachant en tirer parti, comme l'accès direct à des documents ou actions associées à une application.

### Gestion de la JumpList

Par défaut, le système présente une liste des derniers documents ouverts : [Fig.1] Mais il n'y a aucune raison de se limiter à cette seule catégorie. Vous pourriez très bien classer vos documents en termes d'importance : [Fig.2]





Pour créer une catégorie, voici la marche à suivre :

1. Il faut d'abord positionner l'identifiant de l'application. Cet identifiant permet au système d'être certain que toutes les instances de votre application se réfèrent bien à la même application, quelle que soit la façon dont elle a été lancée (explorateur, lien, etc.). Une fois l'identifiant positionné, une référence à la liste peut alors être récupérée.

```
private const string appId = "TaskbarDemo";
Taskbar.AppId = appId;
jumpList = Taskbar.JumpList;
```

2. Il faut ensuite déclarer une nouvelle catégorie de type *CustomCategory* (par opposition aux catégories standard comme « Récent »).

```
private CustomCategory category1 = new CustomCategory("High");
jumpList.CustomCategories.Add(category1);
```

3. Une fois créée, on peut alors ajouter des documents à la catégorie.

```
category1.JumpListItems.Add(new JumpListItem(path));
```

4. Enfin, il faut mettre à jour la liste.

```
Taskbar.JumpList.RefreshTaskbarList();
```

De la même façon que vous pouvez gérer l'accès direct à des documents, vous pouvez aussi gérer le lancement de tâches : [Fig.3] L'API ne diffère guère :

1. L'ajout d'une tâche se fait par ajout d'un objet *JumpListLink* via *UserTasks.Add*.

```
// Path to Windows system folder
string systemFolder =
Environment.GetFolderPath(Environment.SpecialFolder.System);

// Add our user tasks
jumpList.UserTasks.Add(new JumpListLink
{
    Title = "Open Notepad",
    Path = Path.Combine(systemFolder, "notepad.exe"),
    IconReference =
new IconReference(Path.Combine(systemFolder, "notepad.exe"), 0)
});
```

2. L'ajout d'une tâche se fait par ajout d'un objet *JumpListSeparator* via *UserTasks.Add*.

```
jumpList.UserTasks.Add(new JumpListSeparator());
```

3. Et quoi qu'il arrive, il faut mettre à jour la liste.

```
Taskbar.JumpList.RefreshTaskbarList();
```

## Gestion du statut

L'une des premières applications à avoir tiré parti du système de notification de la *TaskBar* fut Live Messenger. Selon que vous soyez connecté ou non, l'icône associée à l'application expose un visuel différent :



Ce système de notification est basé sur la juxtaposition d'une icône au-dessus de l'icône de l'application (d'où l'appellation d'« Overlay Icon »). Pour rendre ce type d'effet dans vos applications, vous devez :

1. Intégrer une icône en tant que ressource dans votre application.
2. Positionner cette icône à l'exécution.

```
Taskbar.OverlayImage =
new OverlayImage(TaskbarDemo.Properties.Resources.Yellow, "Yellow");
```

## Barre de progression

Lorsque vous copiez des fichiers avec l'explorateur ou téléchargez quelque chose avec votre navigateur, l'application notifie l'avancement de l'opération via une barre de progression affichée en surimpression de l'icône de l'application :

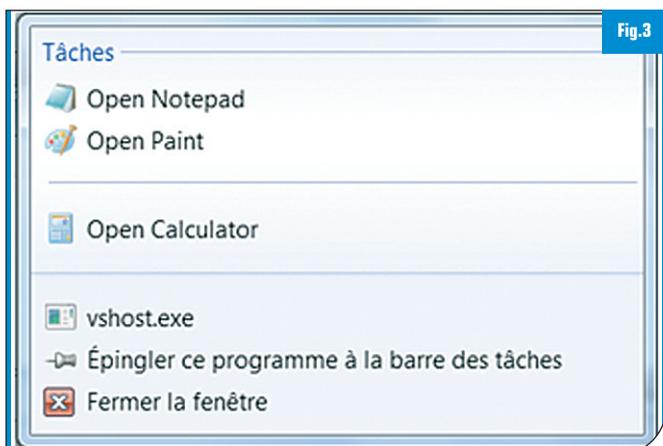


Vous pouvez gérer le même effet dans vos applications de la manière suivante via les propriétés *Taskbar.ProgressBar* ou *Taskbar.MultipleViewProgressBar*. La principale différence entre ces deux propriétés réside principalement dans le suivi de l'avancement pour une ou plusieurs fenêtres, même si, quel que soit le cas, il n'y aura jamais qu'une seule barre affichée à un instant donné.

L'état de la barre de progression peut prendre plusieurs valeurs :

- **NoProgress**, pas de barre affichée.
- **Indeterminate**, pas de valeur précise.
- **Normal**, une barre de progression normale, avec une couleur verte.
- **Error**, la couleur de la barre passe au rouge.
- **Paused**, la couleur de la barre passe au jaune.

Et pour afficher la barre avec une certaine progression, vous positionnez :



# dossier \ windows 7

1. Son état.

```
Taskbar.ProgressBar.State = TaskbarButtonProgressState.Normal;
```

2. Et la valeur de la progression (ici 70 %).

```
Taskbar.ProgressBar.CurrentValue = 70;
```

## Gestion des aperçus

Dans la continuité de Vista, la barre de tâches permet de pré visualiser le contenu de chaque fenêtre : [Fig.4]

Internet Explorer étend le système en affichant une vue par onglet ouvert. Vous pouvez rendre la même expérience grâce à la propriété *TabbedThumbnail*. Elle vous permet d'associer une vue à une contrôle et d'en faire la vue active :

```
TabbedThumbnailPreview preview =
    new TabbedThumbnailPreview(this.Handle, myControl.Handle);
Taskbar.TabbedThumbnail.AddThumbnailPreview(preview);

Taskbar.TabbedThumbnail.SetActiveTab(myControl);
```

Ces API possèdent leurs équivalents WPF. Une fois la vue affichée, vous pouvez réagir de deux façons à la manipulation de la vue par l'utilisateur :

1. En répondant à un événement standard, comme *Taskbar.TabbedThumbnail.TabbedThumbnailActivated*.
2. Ou en ajoutant des commandes associées à votre vue, comme le fait Windows Media Player. [Fig.5]

La définition de telles commandes passe par :

1. La définition d'une instance de *ThumbnailButton* avec une ressource associée pour le visuel.

```
private ThumbnailButton thumbButtonBack;
thumbButtonBack =
    new ThumbnailButton(Properties.Resources.prevArrow, "Back");
thumbButtonBack.Click += new EventHandler(thumbButtonBack_Click);
```

2. L'association des boutons à une instance de contrôle Windows Forms ou WPF.

```
Taskbar.ThumbnailToolbars.AddButtons(newTab.Handle, thumbButtonBack);
```

Enfin, l'API *GrabWindowBitmap*, fonctionnant elle aussi avec des contrôles Windows Forms ou WPF, vous sera utile si vous avez besoin de rafraîchir une vue :

```
TabbedThumbnailPreview preview =
    Taskbar.TabbedThumbnail.GetThumbnailPreview(myControl);
Bitmap bitmap =
    ScreenCapture.GrabWindowBitmap(tabPage.Handle, tabPage.Size);
preview.SetImage(bitmap);
```

## Les bibliothèques

Avec Windows 7, une nouvelle approche dans la gestion des fichiers fait son apparition. Jusque-là, le système proposait pour chaque utilisateur des espaces de stockages prédéfinis comme « Mes Documents » ou « Mes Images ». Chacun de ces espaces était identifié de manière unique (par une Guid en pratique) et automatiquement indexé. Mais l'expérience a montré que cette façon de faire avait sa limite : la façon dont les utilisateurs structuraient le stockage de leurs données était plus complexe que cela, le résultat étant que les utilisateurs finissaient par stocker des données qu'ils avaient du mal ensuite à retrouver (car non indexées).

Les bibliothèques tentent de répondre à cette problématique en offrant une vue du système de fichier en lieu et place de répertoires prédéfinis. Ainsi, la bibliothèque Documents regroupe par défaut tous les documents de l'utilisateur (%USERPROFILE%\Documents) et ceux du profil Public : [Fig.6]

Lorsque l'utilisateur passe par l'explorateur, une des boîtes de dialogue standard ou fait une recherche, il n'a pas besoin de savoir si le document qu'il recherche est physiquement situé dans l'un de ces répertoires, ou dans un répertoire qu'il aurait pu ajouter à la bibliothèque. Il peut directement travailler sur cette vue de l'ensemble des documents.

Une application peut travailler de plusieurs façons avec les bibliothèques. La plus simple est de se reposer sur les boîtes de dialogue standard introduites avec Vista : [Fig.7]

Elles gèrent à la fois les bibliothèques et les emplacements standard. Le souci ici est d'utiliser la bonne API. En effet, pour des raisons de compatibilité ascendante, le système embarque deux versions des boîtes standard : une compatible XP et une compatible Vista/7. Si vous utilisez l'espace de nom *Microsoft.Win32* (comme *Microsoft.Win32.SaveFileDialog*), vous aurez une boîte de dialogue compatible XP. Bien que les bibliothèques soient visibles, elles ne sont pas exploitées au mieux. A la place, mieux vaut recourir à l'espace de nom *System.Windows.Forms* (comme *System.Windows.Forms.SaveFileDialog*), même dans vos applications WPF. Mais si vous souhaitez une intégration plus forte, le namespace *Microsoft.WindowsAPICodePack.Shell* est fait pour vous. Ainsi, pour lister toutes les bibliothèques de l'utilisateur, vous pouvez utiliser la collection *KnownFolders.Libraries* :



Fig.4



Fig.5



```
IEnumerable<ShellObject> libraries = from folder in Known
Folders.Libraries
select folder;
```

De la même façon, pour parcourir tous les répertoires « connus » de l'utilisateur, vous pouvez utiliser la collection *KnownFolders.All*. La classe statique *KnownFolder* expose ainsi l'essentiel des répertoires couramment utilisés dans un contexte utilisateur : documents, images, applications 32 bits et 64 bits, profil itinérant, fichiers temporaires, etc...

Enfin, vous trouverez aussi dans cet espace de noms des contrôles Windows Form (*Microsoft.WindowsAPICodePack.Shell.ExplorerBrowser*) et WPF (*Microsoft.WindowsAPICodePack.Shell.Presentation.ExplorerBrowser*) vous permettant d'intégrer des vues dans vos applications : [Fig.8]

### Les task dialogs

Les Task Dialogs, disponibles depuis Windows Vista, ont été conçues pour permettre aux applications d'afficher des choix riches et multiples de manière consistante. Tout comme les fenêtres d'ouverture ou de sauvegarde de fichiers, les TaskDialogs fournissent aux utilisateurs une interface à la fois familière et personnalisable pour des opérations récurrentes. L'exemple type de leur utilisation est la fenêtre d'avertissement de l'explorateur de fichier [Fig.9].

Bien plus personnalisable qu'une MessageBox classique, la TaskDialog sépare visuellement et interactivement les contextes de choix. Son wrapper WACP dispose de nombreuses propriétés aux noms explicites, résumés dans l'exemple suivant : [Fig.10]

On note que les boutons de choix sont des objets à part entière et sont de type *TaskDialogCommandLink*. Cela leur permet d'être personnalisables à la fois dans leur apparence mais également par leur comportement. Il est ainsi possible d'ajouter un nombre variable de boutons avec la méthode *TaskDialog.Add*, et d'attacher un délégué à l'évènement *TaskDialogCommandLink.Click* de chacun d'eux :

```
TaskDialogCommandLink monTaskDialogCommandLink = new Task
DialogCommandLink("Un", "Premier TaskDialogCommandLink");
monTaskDialogCommandLink.Click += new EventHandler(tdcLink
_Click);
demoTaskDialog.Controls.Add(monTaskDialogCommandLink);
...
static void tdcLink_Click(object sender, EventArgs e)
{
```

```
// Send feedback button
...
```

Une TaskDialog n'a pas forcément un contenu aussi riche que décrit précédemment. Il est aussi possible d'utiliser cette API en lieu et place d'une MessageBox tout en bénéficiant d'un rendu visuel légèrement amélioré, avec notamment la possibilité d'afficher un titre (*InstructionText*) en plus de celui de la fenêtre : [Fig.11]

```
TaskDialog simpleTaskDialog = new TaskDialog();
simpleTaskDialog.Icon = TaskDialogStandardIcon.Warning;
simpleTaskDialog.StandardButtons = TaskDialogStandardButtons
.OkCancel
| TaskDialogStandardButtons.YesNoCancel
| TaskDialogStandardButtons.Close;
simpleTaskDialog.InstructionText = "Ceci est l'Instruction
Text";
simpleTaskDialog.Caption = title = "Ici le contenu de la propriété
Caption";
simpleTaskDialog.Text = "Ici le contenu de la propriété Text.
La valeur renvoyée"
+ "par la sélection d'un des boutons est de type Task
DialogResult.";
TaskDialogResult res = simpleTaskDialog.Show();
```

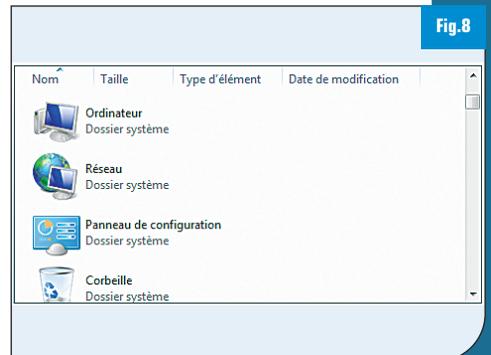
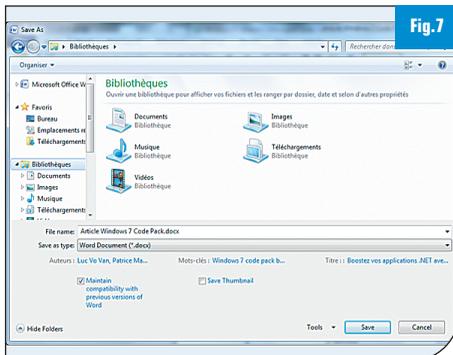
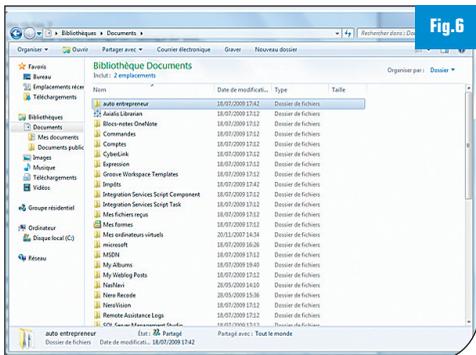
Notons que la version de WACP disponible lors de la rédaction de cet article (0.90) ne permet pas encore l'utilisation d'icônes et de titres dans les *TaskDialogButtons*.

### L'Application Recovery

Disponible depuis Vista, cette fonctionnalité permet aux applications de s'enregistrer auprès de l'OS afin qu'en cas de crash, elles soient redémarrées automatiquement en retrouvant leur dernier état stable. C'est par exemple le cas des applications de Microsoft Office qui, lorsqu'elles subissent un arrêt inattendu, redémarrent automatiquement et permettent à l'utilisateur de récupérer des données qu'il n'a pas nécessairement eu le temps de sauvegarder. Depuis Vista, la fenêtre suivante est affichée lorsqu'une application cesse de répondre : [Fig.12]

Le fonctionnement de ce système est le suivant :

1. Au démarrage, l'application s'enregistre auprès de l'OS par un appel à la méthode *ApplicationRestartRecoveryManager.RegisterForApplicationRestart*. On fournira à cette méthode un argument de ligne de commande qui sera utilisé par l'OS en cas de redémarrage de l'application, ainsi que des



# dossier \ windows 7

options permettant de personnaliser les cas de redémarrage (par exemple, `RestartRestrictions.NotOnReboot`, ou encore `RestartRestrictions.NotOnPatch` pour ne pas redémarrer suite à l'application d'un patch).

```
ApplicationRestartRecoveryManager.RegisterForApplicationRestart(
    new RestartSettings("/restart", RestartRestrictions.NotOnReboot |
        RestartRestrictions.NotOnPatch));
```

2. Toujours au démarrage, l'application s'enregistre pour les éventuelles opérations de récupération de données. L'appel à `ApplicationRestartRecoveryManager.RegisterForApplicationRecovery` permet de préciser un délégué et un paramètre qui seront appelés dans le cas d'un redémarrage automatique. Le code de cette méthode est chargé d'enregistrer (récupérer) les informations de la session ayant crashé, puis d'appeler `ApplicationRestartRecoveryManager.ApplicationRecoveryFinished(bool)` pour indiquer la fin de l'opération de récupération (`true` indiquant une récupération réussie). On enregistre typiquement le résultat de cette opération dans un fichier temporaire dont le contenu sera utilisé lors du redémarrage de l'application.

La récupération des informations pouvant durer un certain temps, un appel périodique à `ApplicationRestartRecoveryManager.ApplicationRecoveryInProgress()` est nécessaire afin de notifier l'OS que l'opération de récupération n'est pas à son tour bloquée. Cette méthode renvoie également un booléen indiquant si l'utilisateur a annulé le redémarrage en cliquant sur le bouton d'annulation.

3. L'application est ensuite redémarrée par l'OS et doit vérifier le premier paramètre qui lui est passé, s'il existe. S'il s'agit du paramètre précisé à l'étape 1, on peut alors supposer qu'il s'agit d'un redémarrage automatique. Dans ce cas, on lit les informations récupérées lors de l'étape 2 pour rétablir l'état de l'application : [Fig.13]

Afin d'éviter des redémarrages en boucle, l'*Application Recovery* ne fonctionne que pour des applications démarrées depuis au moins une minute.

## Gestion du réseau

Le WACP ajoute aux nombreuses capacités réseau du framework .NET quelques nouvelles fonctionnalités très pratiques liées à la connectivité. Ces nouvelles fonctionnalités permettent d'obtenir de nombreuses informations que l'on a l'habitude de voir au niveau de l'interface utilisateur de l'OS : les différents réseaux auxquels l'ordinateur s'est déjà connecté, leurs catégories (public, privé, authentification), leur dernière date d'ac-

cess ou encore s'il s'agit de réseaux de domaine. Cependant, la fonctionnalité la plus intéressante concerne les informations sur les connexions en cours. Il est ainsi très facile de déterminer les réseaux auxquels l'ordinateur est actuellement connecté, par quelles interfaces matérielles, si ces connexions permettent l'IPv6, ou encore s'ils donnent accès à Internet.

La classe statique `NetworkListManager` est le point d'entrée pour ces données. Deux de ses propriétés informent sur l'état global de connexion : `NetworkListManager.IsConnected` permet de déterminer si l'ordinateur est actuellement connecté à au moins un réseau, tandis que `NetworkListManager.IsConnectedToInternet` permettra de déterminer si au moins une des connexions réseau actives permet d'accéder à Internet. Cette dernière propriété peut s'avérer particulièrement intéressante lorsque l'on souhaite éviter d'effectuer des requêtes qui n'auront aucune chance d'aboutir et pouvant entraîner un délai avant de renvoyer une erreur de *timeout*.

Il est aussi possible d'obtenir différents types de liste de réseaux en appelant la méthode `NetworkListManager.GetNetworks(NetworkConnectivityLevels level)`, qui en fonction de son paramètre renverra tous les réseaux auxquels la machine s'est déjà connectée, soit les réseaux déconnectés, soit les réseaux connectés. La liste renvoyée contient des instances d'objets `Network` portant chacun de nombreuses informations dont un nom, un état de connectivité à Internet, ou encore un type (Professionnel, Résidentiel, Public).

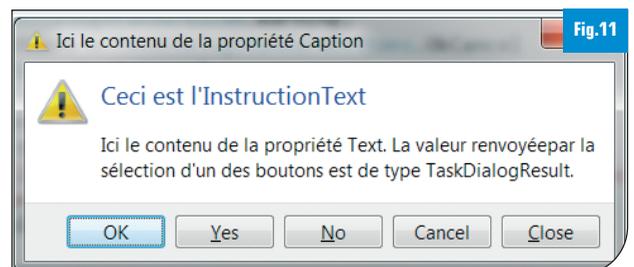
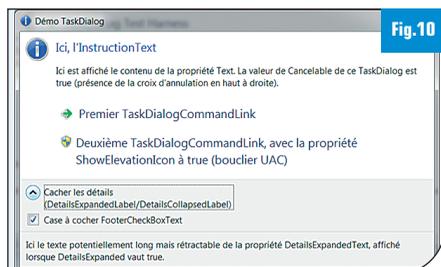
Par exemple, une simple requête LINQ permettra de déterminer les connexions en cours ayant un accès à Internet et étant configurées comme privées :

```
var resConn = from net in NetworkListManager.GetNetworks(
    NetworkConnectivityLevels.Connected)
    where net.IsConnectedToInternet && net.Category
    == NetworkCategory.Private
    select net;
```

Des appels réguliers à `NetworkListManager.GetNetworks` permettent ainsi, par exemple, de déterminer si des connexions réseau ont été interrompues ou encore de s'assurer du bon fonctionnement des différentes interfaces réseaux et de réagir en fonction de leur état (levée d'alerte, mise en attente de certaines tâches).

## L'API de gestion d'énergie

L'adaptation du comportement d'une application en fonction de la source d'énergie de l'ordinateur devient une fonctionnalité de plus en plus importante. En effet, les préoccupations environnementales et l'impressionnante popularité des ordinateurs portables ont fait de la gestion d'énergie un des sujets fétiches de ces dernières années. Les mesures pouvant être prises par les logiciels



# Vitesse, mémoire, sécurité.



## Les serveurs dédiés PRO offrent d'énormes performances.

REFLEXION/FAITE - Conditions générales de vente sur le site [www.amen.fr](http://www.amen.fr); AMEN RCS PARIS : B 421 527 797.

À partir de  
**99€** HT/mois\*  
Satisfait ou remboursé\*\*

ASSISTANCE TECHNIQUE  
PAR TELEPHONE 7J/7<sup>(1)</sup>

AVEC LES PROCESSEURS AMD OPTERON™ QUADRIprocesseurs, VOUS disposez d'une incroyable puissance de traitement et d'une énorme capacité de mémoire. Vous pouvez ainsi héberger vos applications métiers les plus exigeantes ou des sites web à fort trafic en toute sécurité.

- AMD Opteron™ 1352, 1354 ou 1356
- De 4x2,1 GHz à 4x2,3 GHz
- De 4 à 8 Go de RAM
- Disques durs de 2x500 Go à 2x1 To
- Raid 1 Hardware
- 2 à 6 adresses IP
- Trafic mensuel illimité
- OS 64 bits : Debian 4, Ubuntu 8.04 LTS, CentOS 5 ou Windows Server 2008
- Interface Plesk de 30 à 300 domaines
- Sauvegarde FTP : de 40 Go à 80 Go
- Services Inclus : Reboot, Restore et Recovery
- Assistance technique par téléphone 7J/7<sup>(1)</sup>

Amen et Dada : 1,4 million de domaines gérés et plus de 500 000 sites hébergés.



# Amen

A DADA COMPANY

0 800 74 09 35 (Numéro gratuit) - [www.amen.fr](http://www.amen.fr)

NOMS DE DOMAINE - EMAIL - HÉBERGEMENT - CRÉATION DE SITE - E-COMMERCE - RÉFÉRENCIEMENT

\*Pour un engagement annuel. \*\* Sous 10 jours.  
(1) 06 99 70 90 01, coût de la communication pour l'assistance technique : 1,35 € l'appel + 0,34 € la minute.

# dossier \ windows 7

pour permettre d'économiser l'énergie sont nombreuses, par exemple :

- Réduire les effets graphiques en mode économie d'énergie ou lorsque la machine fonctionne sur batterie
- Proposer de baisser la luminosité d'un écran CRT
- Eteindre ou rendre plus rare l'exécution de certaines actions (indexation, calcul en tâche de fond...)
- Conseiller l'utilisateur en fonction de la charge restante ou de la source de courant (sauvegarde, avertissement de forte consommation pour certaines applications)
- Temps de charge restant ou absence d'onduleur pour l'exécution de tâches critiques.

Des exemples concrets sont présents dans Windows qui, en mode économie d'énergie, désactive certains effets graphiques et réduit par défaut la luminosité de l'écran de portables. Windows Presentation Foundation (WPF) désactive également par défaut certaines animations subtiles (par exemple la légère lueur des boutons) lorsque l'ordinateur fonctionne sur batterie.

La classe statique `PowerManager` du WACP permet un accès aisé aux différentes informations de gestion d'énergie remontées par la machine au système d'exploitation. Cette classe propose ainsi des événements et des propriétés liées à la gestion d'énergie, dont voici quelques exemples d'utilisation.

L'évènement `PowerManager.PowerSourceChanged` permet de savoir que la source principale d'alimentation a changé. Sur le déclenchement de cet évènement, une simple vérification de la valeur de `PowerManager.PowerSource` permettra de déterminer si l'ordinateur fonctionne sur secteur ou sur batterie.

```
PowerManager.PowerSourceChanged += delegate
{
    switch (PowerManager.PowerSource)
    {
        case PowerSource.AC:
            // On est sur secteur
            break;
        case PowerSource.Battery:
        case PowerSource.Ups:
            // On est sur batterie ou onduleur
            break;
    }
};
```

L'évènement `PowerManager.SystemBusyChanged` est déclenché par le système pour avertir les programmes que la machine n'entrera pas en mode *idle* (basse consommation) dans les secondes à venir. Il est alors conseillé de profiter de ces instants d'activité de l'ordinateur pour exécuter des tâches de fond ou en attente.

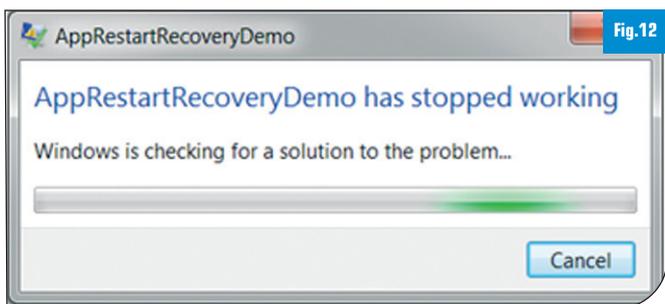


Fig.12

Les propriétés `PowerManager.BatteryLifePercent` et `BatteryState.EstimatedTimeRemaining` (obtenu après un appel à `PowerManager.GetCurrentBatteryState()`) permettent respectivement de connaître le pourcentage de batterie chargée, et le temps estimé avant qu'elle ne se vide.

La classe `PowerManager` donne aussi accès à de nombreuses autres informations et événements que nous ne décrirons pas en détail ici, mais dont les noms bien choisis vous permettront de deviner facilement l'utilité.

## Support des versions antérieures

Comme pour la majorité des développeurs, l'idée de restreindre votre application à un OS en particulier n'est pas très agréable. Il serait dommage de réserver votre application aux seuls utilisateurs de Windows 7 parce que Windows XP ou Windows Vista ne disposent pas de la barre de progression dans l'icône de la barre des tâches !

C'est la raison pour laquelle la classe `Microsoft.WindowsAPICodePack.CoreHelpers` vous permettra de détecter la version de Windows sur laquelle s'exécute votre application. Vous pourrez ainsi adapter votre code et éviter des dysfonctionnements. Prenons par exemple une application pouvant mettre à jour une barre de progression, si elle s'exécute sur un Windows 7, elle pourrait en profiter pour mettre à jour l'icône de l'application.

```
myProgressBar.Value++;
if (CoreHelpers.RunningOnWin7)
    Taskbar.MultipleViewProgressBar.SetCurrentValue(this, myProgressBar.Value);
```

Dans le cas plus restrictif où votre application requiert une version d'OS minimale, vous pourrez utiliser les méthodes `ThrowIfNot` :

```
CoreHelpers.ThrowIfNotVista();
TaskDialog.Show("TaskDialog vraiment nécessaire ici!");
```

## Conclusion

L'amélioration de l'expérience utilisateur est un des axes majeurs du nouvel OS de Microsoft, et le Windows API Code Pack donne aux développeurs .NET un accès privilégié au fruit de ce travail. Nous espérons que les différentes fonctionnalités décrites dans cet article vous auront permis de découvrir agréablement comment utiliser cette librairie afin d'améliorer l'expérience de vos utilisateurs à la fois sur vos applications existantes, et celles à venir !

Pour en savoir plus :

<http://code.msdn.microsoft.com/WindowsAPICodePack>

■ Patrice Manac'h & Luc Vo Van

Consultants Microsoft France

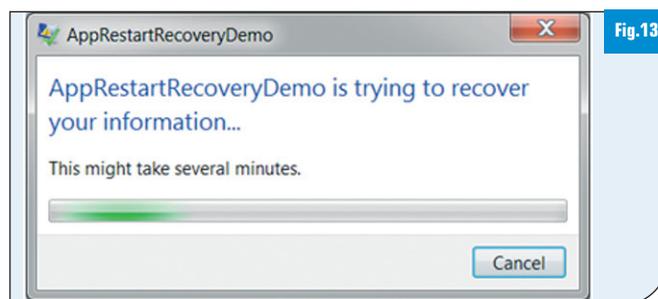


Fig.13



# Utiliser le multi-touch dans ses applications



Une des nouvelles caractéristiques de Windows 7 est le support d'une interface tactile Multi-Touch. Pour développer une interface possédant de telles fonctionnalités, la firme de Redmond met à notre disposition Visual Studio 2010 et le Net 4.0.

Des technologies découlant de .Net comme WPF ou Silverlight ont donc évolué. Pour rappel, WPF est une des technologies de développement d'application du Framework .NET. Elle fournit également une adaptation de développement d'application web : Silverlight. Côté software, les prérequis pour développer une application multi-touch sont Windows 7, Visual Studio 2010 et le Training Kit de Visual Studio 2010. Côté matériel, il vous faudra vous munir d'un écran multi-touch, c'est-à-dire capable de reconnaître les contacts simultanés.

## Présentation du multi-touch sous Windows 7

En ajoutant la fonctionnalité tactile, Microsoft a enrichi l'expérience utilisateur. En effet, de nouvelles possibilités de navigation s'offrent à l'utilisateur. L'exemple le plus simple est le visionnement de photos. Vous pouvez maintenant manipuler vos photos en choisissant de les faire défiler plus ou moins rapidement ou effectuer un zoom ou une rotation de celles-ci en agissant directement avec votre ordinateur. [Fig.1]

Microsoft a également su répondre aux problématiques liées à l'adaptation d'un système d'exploitation au multi-touch. Par exemple, comment effectuer un « clic droit » avec le doigt ? Sous Windows 7, il suffira de maintenir enfoncé le doigt 2 secondes ou alors de toucher l'écran avec deux doigts. Ainsi, de nombreux outils ont été optimisés pour l'expérience multi-touch comme la barre des tâches, la visionneuse XPS, Windows Photo Viewer, Internet Explorer 8...

Essayons de comprendre comment fonctionne la plate-forme tactile multi-touch Windows 7. Chaque entrée tactile à un ou deux doigts se traduit par un geste que l'utilisateur effectue. A partir du moment où Windows 7 détecte une entrée, il envoie directement à votre application un message de geste appelé WM\_GESTURE

contenant toutes les informations le concernant. Les gestes pris en charge par Windows 7 sont les suivants :

- Zoom
- Rotation
- Pression
- Geste panoramique à un ou deux doigts
- Toucher à un ou deux doigts

L'utilisation de ces gestes doit être pour le développeur le moyen d'améliorer et d'optimiser l'expérience utilisateur pour un fonctionnement multi-touch. De nombreuses entreprises ont déjà commencé à développer des applications assez impressionnantes en s'appuyant sur du matériel évolué. Par exemple, une sphère tactile, un bureau entièrement tactile ou encore une table de salon tactile permettant de gérer tout son matériel électroménager comme l'aspirateur, le home cinema, la TV et même la luminosité de la pièce. Toutes ces adaptations nous montrent que les possibilités de développement liées au multi-touch s'appliquent à la fois aux applications ludiques et aux applications professionnelles. La révolution tactile est en marche !

## WPF 4 : l'armature

L'arrivée du Framework 4 entraîne celle de WPF 4. Ce dernier va être une très bonne alternative pour pallier les difficultés rencontrées dans le développement d'application multi-touch avec le précédent Framework ( 3.5 ). En effet, parmi les nouveautés de WPF 4, il y a l'intégration de base des événements de manipulation qui sont implémentés sur les UIElements. Aussi, un nouveau contrôle fait son apparition ( le Scatter View ) et la mise à jour pour le support du multi-touch du ScrollViewer. [Fig.2]

Effectivement, l'interface tactile n'est pas nouvelle pour Windows puisque Vista intégrait déjà auparavant de nombreuses fonction-



Fig.1



Fig.2

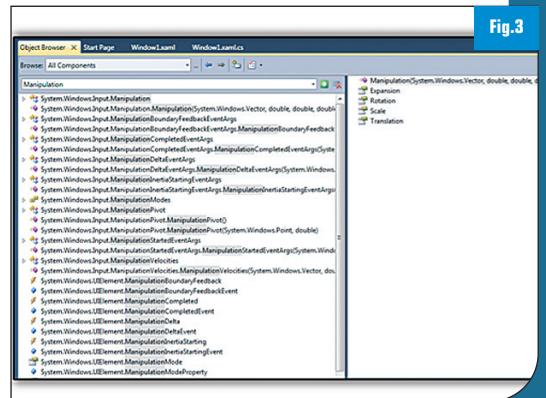


Fig.3

# dossier \ windows 7

nalités multi-touch. Le développement va s'effectuer alors avec WPF 2 et on retrouvait déjà certains événements utilisés avec le Framework 4 comme *StylusEnter*, *StylusMove*, *StylusUp*. Preuve que l'innovation tactile a déjà été pensée sous Vista mais n'avait pas su tirer le meilleur parti de cette innovation. Windows 7 et l'arrivée du WPF 4 apportent de réelles opportunités pour faire de nos applications des interfaces beaucoup plus riches.

## Découverte du contrôle ScatterView

Pour commencer cette découverte du Multi-Touch nous allons faire une analyse rapide et simple du contrôle mis à notre disposition par le Framework 4. Le Scatter View est un contrôle issu de Surface qui permet un visionnement de photos très « user friendly » avec une interface tactile. Ce contrôle est disponible dans Microsoft.Windows.Controls de base, mais bien entendu il ne faut pas oublier de rajouter la dll du ScatterView disponible dans le Training Kit de Visual Studio 2010. Ainsi, après déclaration du ScatterView et des images, on obtient une interface graphique déjà toute prête tant pour les manipulations que pour la gestion des Zooms, translations, rotations etc. Attention, il ne faut pas oublier de rajouter impérativement l'assembly suivante :

```
<xml ns:= "clr-namespace:Microsoft.Windows.Controls;assembly=ScatterView"
```

Inutile de s'attarder sur ce contrôle, son fonctionnement est identique à celui de Surface.

## Les nouveaux événements

Rentrons plus dans le vif du sujet en créant à la main une sorte de ScatterView grâce, non pas au contrôle prédéfini, mais plutôt par l'intermédiaire des nouveaux événements.

Pour avoir un aperçu de tous ces événements rendez-vous dans l'explorateur d'objet de votre Visual Studio 2010 et rentrez « manipulation » dans la barre de recherche. [Fig.3].

- ManipulationStarted
- ManipulationDelta (certainement le plus utilisé)
- ManipulationStarting
- ManipulationCompleted

Maintenant que vous avez à l'esprit une idée de ces événements, il vous faut bien entendu mettre en place l'interface graphique de votre application en disposant par exemple quelques photos dans un Canvas. N'oubliez surtout pas de nommer votre Canvas (Nous l'avons appelé *canvasContenu*).

Ensuite, vous allez remarquer que dans les attributs d'une image une nouvelle propriété est à notre disposition. « *ManipulationMode* » définissant ainsi le type de multi-touch souhaité (Translation, Rotation, Zoom ...).

Une question se pose alors : comment allons-nous modifier les coordonnées de l'image lors des manipulations ? L'utilisation de matrice nous est alors indispensable et elle s'appliquera à toutes nos images automatiquement car nous n'agissons pas sur chaque image indépendamment mais sur le canvas qui les contient toutes, en établissant aussi la position de cette matrice. Il faut par la suite appliquer la matrice ainsi que sa transformation aux images. Pour cela, nous n'avons qu'à rajouter la propriété permettant cette liaison dans les contrôles image.

```
<Window x:Class="ApplicationTouC#WPF4.Window1"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:i="http://schemas.microsoft.com/winfx/2006/xaml"
Title="Window1">
<Grid>
<Canvas x:Name="canvasContenu">
<Canvas.Resources>
<MatrixTransform x:Key="Matrice">
<MatrixTransform.Matrix>
<Matrix OffsetX="350" OffsetY="350"/>
</MatrixTransform.Matrix>
</MatrixTransform>
</Canvas.Resources>
<StackPanel RenderTransform="{StaticResource Matrice}" Height="100"
Width="100" Background="Black"/>
<Image Source="1.jpg" Width="100" Height="100"
RenderTransform="{StaticResource Matrice}" ManipulationMode="All" />
</Canvas>
</Grid>
</Window>
```

Une fois toute la partie graphique mise en place, nous allons gérer nos événements en code métier. Dans notre exemple, nous utiliserons le C#. Tout d'abord, la déclaration de notre événement que l'on a appelé *ManipulationImage*, aura en paramètre *object sender* comme tout événement mais aussi *ManipulationDeltaEventArgs*. Mais que signifie ce delta ? Et bien il correspond à la valeur par défaut du commutateur, ainsi le message de geste est général, c'est-à-dire que nous n'avons pas de début ni de fin.

Les transformations découlant des manipulations seront déclarées entre les balises de cet événement. Mais dans un premier temps nous allons récupérer les nouvelles coordonnées par rapport à notre canvas, grâce à *GetDeltaManipulation* qui va retourner les changements les plus récents réalisés lors de la dernière manipulation. Puis on récupère par la même occasion l'image manipulée. Celle qui sera donc modifiée sous l'action des doigts de l'utilisateur. Ensuite, nous en déduisons sa matrice afin de pouvoir altérer ses coordonnées et ainsi pouvoir lui affecter et appliquer toutes les transformations subies.

Maintenant, il ne nous reste plus qu'à définir ce que fera chaque transformation. Gardez à l'esprit que les transformations agissent sur la matrice et non sur les propriétés de l'image directement. Il est logique que travailler sur une seule matrice soit impossible avec plusieurs images. On ne peut pas manipuler une image si les transformations sont appliquées à une autre image. Pour résoudre cet inconvénient majeur, rien de plus simple que de créer une nouvelle matrice à chaque nouvelle manipulation étant donné qu'au début nous récupérons les coordonnées qui seront modifiées ou non par une quelconque manipulation antérieure.

```
private void ManipulationImage (object sender, ManipulationDeltaEventArgs e)
{
//On récupère les nouvelles coordonnées par rapport au canvas principal
Manipulation m = e.GetDeltaManipulation(canvasContenu);

//On récupère l'image manipulée
StackPanel nonButtonMan = e.OriginalSource as StackPanel;

//On en déduit sa matrice, on lui appliquera toutes les transformations
var matrice1 = ((MatrixTransform)nonButtonMan.RenderTransform).Matrix;
var orgCenter1 = new Point(nonButtonMan.ActualWidth / 2,
nonButtonMan.ActualHeight / 2);

//Translation
matrice1.Translate(m.Translation.X, m.Translation.Y);
var center1 = matrice1.Transform(orgCenter1);

//Rotation
matrice1.RotateAt(m.Rotation, center1.X, center1.Y);
center1 = matrice1.Transform(orgCenter1);

//Zoom/D ezoom
matrice1.ScaleAt(m.Scale, m.Scale, center1.X, center1.Y);

//Application des modifications
nonButtonMan.RenderTransform = new MatrixTransform(matrice1);
e.Handled = true;
}
```

# WINDEV : VOUS AUSSI, DÉVELOPPEZ 10 FOIS PLUS VITE !



Lille



Paris

Quelques photos du Tour de France WINDEV 14 ( 6 des 11 villes )



Montpellier



Bordeaux



Lyon



Bruxelles



WINDEV : élu «Langage le plus productif du marché»  
[www.pcsoft.fr](http://www.pcsoft.fr)



**COMMENT DÉVELOPPER + VITE ?  
DEMANDEZ  
LE DOSSIER GRATUIT !**

252 pages + DVD + Version Express incluse  
+ 112 Témoignages.  
Tél: 04.67.032.032 ou 01.48.01.48.88  
[info@pcsoft.fr](mailto:info@pcsoft.fr)



## dossier \ windows 7



Fig.4

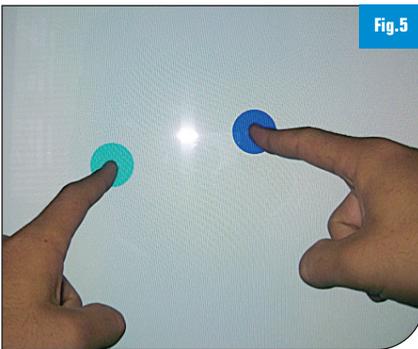


Fig.5



Fig.6

Comme expliqué précédemment, nous allons solliciter *ManipulationDelta*. Néanmoins, vous remarquerez que les événements ainsi que toutes les manipulations ne sont pas fonctionnels. Nous allons rajouter une propriété clé dans notre application qui va permettre de gérer l'évènement manipulation en code métier. Dans la partie XAML de notre Application, nous devons informer que nous allons utiliser les manipulations en la déclarant de la sorte *ManipulationDelta= Image\_ManipulationDelta*. [Fig.4]

```
<Window x:Class="ApplicationTouchWPF4.Window1"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="Window1" ManipulationDelta="ManipulationImage">
```

## Silverlight

Le Framework 4 va nous permettre de réaliser des applications web multitouch. Ces dernières ne sont réalisables qu'avec la version 3 de Silverlight. Cependant, le multi-touch n'est pas du tout le même qu'en WPF 4. Il y a beaucoup moins de possibilités.

### Découverte des outils

Voyons dans un premier temps les outils mis à notre disposition pour Silverlight. L'espace de nom contenant les classes qui nous intéressent est *System.Windows.Input*. Vous trouverez à l'intérieur de celui-ci la classe *Touch*. La classe *Touch* (classe static) est la classe principale, elle sert à traiter l'entrée tactile multipoint du système d'exploitation et déclenche l'évènement *FrameReported*. Arrêtons-nous un instant sur cet évènement. Il s'agit en fait du gestionnaire d'évènements tactiles multipoint. Vous pouvez constater que la gestion des entrées multitouch est totalement différente par rapport à celle utilisée en WPF 4, mais aussi par rapport aux entrées plus classiques en Silverlight comme par exemple le clavier ou la souris. [Fig.5]

D'autres classes sont disponibles dans *System.Windows.Input* et peuvent également être utiles. Nous disposons de :

- *TouchDevice*, qui permet de décrire le type de périphérique qui a produit le point de contact.
- *TouchFrameEventArgs*, qui permet de fournir des données à l'évènement *FrameReported*.
- *TouchPoint*, qui représente un point tactile unique.
- *TouchPointCollection*, qui contient une collection de valeurs *TouchPoint*.

### Exemple d'application

Maintenant que nous savons avec quoi procéder, voyons comment réaliser une application multitouch. Nous voulons concevoir une application simple qui fera apparaître des ellipses lorsqu'un ou plusieurs doigts toucheront l'écran. Ces ellipses devront ensuite

suivre le doigt de l'utilisateur. Commençons par la conception de l'IHM de notre application. Nous avons simplement créé deux ellipses. Nous avons défini leur visibilité à « collapsed ». Ensuite, on s'abonne à notre évènement *Touch.FrameReported* dans le code behind.

```
public Page()
{
    InitializeComponent();
    Touch.FrameReported += new TouchFrameEventHandler(Touch_FrameReported);
}
```

Puis, on déclare notre méthode *Touch\_FrameReported*.

```
void Touch_FrameReported(object sender, TouchFrameEventArgs e)
{
    ellipse_1.Visibility = System.Windows.Visibility.Collapsed;
    ellipse_2.Visibility = System.Windows.Visibility.Collapsed;
    //On récupère les différents points de contact
    TouchPointCollection points = e.GetTouchPoints(null);
    //Le point principal
    TouchPoint primaryPoint = e.GetPrimaryTouchPoint(null);

    if (primaryPoint != null)
    {
        //Désactivation du simple click
        if (primaryPoint.Action == TouchAction.Down)
            e.SuspendMousePromotionUntilTouchUp();
        ellipse_1.Visibility = System.Windows.Visibility.Visible;
        Canvas.SetLeft(ellipse_1, primaryPoint.Position.X - 50);
        Canvas.SetTop(ellipse_1, primaryPoint.Position.Y - 50);

        if (points.Count() == 2)
        {
            ellipse_2.Visibility = System.Windows.Visibility.Visible;
            Canvas.SetLeft(ellipse_2, points[1].Position.X - 50);
            Canvas.SetTop(ellipse_2, points[1].Position.Y - 50);
        }
    }
}
```

*Note : Le gestionnaire *Touch.FrameReported* est basé sur le délégué *TouchFrameEventHandler*. La méthode *SuspendMousePromotionUntilTouchUp* permet d'interrompre tout évènement lié à la souris.*

## CONCLUSION

Tout au long de cet article, vous avez sûrement pu constater toute la puissance de développement de la plate-forme tactile de Windows 7. Surtout quand on se penche sur des technologies nouvelles comme WPF 4 avec des manipulations semblables aux gestes, avec des mouvements lisses permis par l'inertie. L'inertie est un sujet très intéressant mais aussi très complexe, ce qui pourrait faire l'objet d'un prochain article. La maniabilité du multi-touch est sûrement encore limitée à l'heure actuelle, mais nous ne serions pas surpris de voir une telle technologie prendre son essor dans les mois à venir. [Fig.6]

■ Thibault Laurens & Chauvin Bastien

■ Julien Dollon

Consultant/Formateur .NET

MVP | Consultant/Formateur Exakis/I'FORM | Président Dotnet-France



# Ecrire des services "déclenchés"

L'un des objectifs fixés aux équipes de développement de Windows 7 fut de répondre aux critiques sur la performance et la réactivité du système. C'est un sujet assez vaste qui couvre aussi bien le démarrage de la machine que l'ouverture d'une session ou le fait de basculer entre deux applications.

Cet article se propose d'aborder une nouvelle fonctionnalité permettant d'alléger la phase de démarrage sans sacrifier aux fonctionnalités : **les services déclenchés par un événement**. Jusqu'à Windows XP, un service était soit paramétré pour démarrer automatiquement, ce qu'il faisait alors pendant le démarrage de la machine, soit manuellement, soit il était désactivé. Avec Vista est apparue une nouvelle option : le démarrage automatique avec délais. Pour prendre un exemple, sur mon poste, j'ai une instance de SQL Server dont je me sers souvent, j'ai donc besoin qu'elle démarre automatiquement. Par contre, lorsque je redémarre ma machine, je n'ai pas forcément besoin que ce service soit disponible tout de suite. Il peut donc démarrer avec un peu de retard, laissant ainsi des services plus prioritaires démarrer tout de suite. En théorie, cela peut donc soulager la phase de démarrage. L'idée a été ici de se dire que certains services ne devraient démarrer automatiquement que sous certaines conditions. Par exemple, imaginons un service chargé de récupérer des mises à jour d'un produit (signatures anti-virus, patches, etc) depuis le site d'un éditeur. Le démarrer si la machine n'est pas connectée ne sert à rien, sauf à consommer des ressources. Par contre, dès que la connexion est établie, il faut que le service démarre tout seul, sans action de l'utilisateur. C'est à ce type de problématique que répond la fonction de service déclenchée par un événement. Et il y a un bonus en prime : en ne faisant fonctionner que ce qui est nécessaire, la consommation électrique et des ressources mémoire/processeur de la machine s'en trouve potentiellement améliorée.

## Les principes

Avec Windows 7 et Windows 2008 R2, une nouvelle structure, SERVICE\_TRIGGER, est donc définie afin de pouvoir décrire sous quelles conditions un service doit démarrer ou s'arrêter. Cette structure est décrite sur [http://msdn.microsoft.com/en-us/library/dd405512\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/dd405512(VS.85).aspx) et permet de préciser :

- Le type d'événement à gérer.
- L'action à mener (démarrage ou arrêt).
- D'éventuels paramètres.

Les événements peuvent être de type :

- **Ajout d'un matériel**. Le champ **dwTriggerType** vaut alors SERVICE\_TRIGGER\_TYPE\_DEVICE\_INTERFACE\_ARRIVAL. Le champ **pTriggerSubtype** contient l'identifiant de la classe de périphérique correspondant. Et le champ **pDataItems** pointe vers le numéro d'identification du matériel.
- **Connectivité** : Le champ **dwTriggerType** vaut alors SERVICE\_TRIGGER\_TYPE\_IP\_ADDRESS\_AVAILABILITY.

Le champ **pTriggerSubtype** permet de spécifier si l'action doit être lancée quand la machine vient d'acquies sa première adresse IP

ou d'être déconnectée de sa dernière adresse.

- **Domaine** : Le champ **dwTriggerType** vaut alors SERVICE\_TRIGGER\_TYPE\_DOMAIN\_JOIN. Le champ **pTriggerSubtype** permet de spécifier si l'action doit être lancée quand la machine vient se rejoindre un domaine ou de le quitter.
- **Pare-feu** : Le champ **dwTriggerType** vaut alors SERVICE\_TRIGGER\_TYPE\_FIREWALL\_PORT\_EVENT. Le champ **pTriggerSubtype** permet de spécifier si l'action doit être lancée quand un port vient d'être ouvert ou fermé. Et le champ **pDataItems** pointe vers le port ou l'application à surveiller.
- **Politique utilisateur ou machine** : Le champ **dwTriggerType** vaut alors SERVICE\_TRIGGER\_TYPE\_GROUP\_POLICY. Le champ **pTriggerSubtype** permet de spécifier si l'action doit être lancée quand le changement est de portée machine ou utilisateur.
- **Sur mesure** : Le champ **dwTriggerType** vaut alors SERVICE\_TRIGGER\_TYPE\_CUSTOM. Les champs **pTriggerSubtype** et **pDataItems** permettent alors de spécifier sur quel événement ETW l'action doit être déclenchée.

## La pratique

Pour coder un tel service, il vous faut des éléments assez classiques :

- Un projet de type service Windows (c'est bien le moins). [Fig.1]
- Un composant *installer*. Ce composant va vous permettre d'intervenir après l'installation du service afin de changer ses paramètres et de spécifier sur quel événement le service doit être démarré ou stoppé. [Fig.2]

Une fois cette classe créée, abonnez-vous sur l'événement **AfterInstall** de votre instance d'objet **ServiceInstaller**. C'est dans cet événement que vous allez procéder à la configuration.

Commencez par vous assurer que le code est bien lancé sur un noyau 6.1 ou ultérieur (Windows 7 ou Windows 2008 R2) :

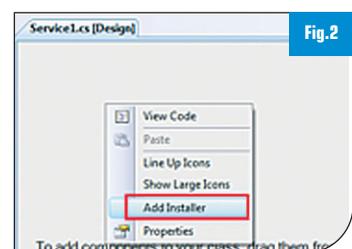


Fig.2

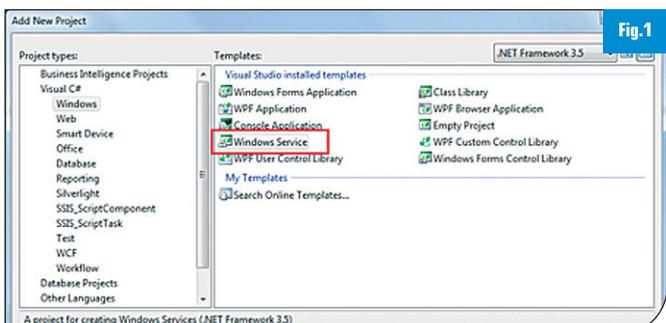


Fig.1

# dossier \ windows 7

```
if (Environment.OSVersion.Version.Major > 6 ||
    Environment.OSVersion.Version.Major == 6 &&
    Environment.OSVersion.Version.Minor >= 1)
```

Une fois cela fait, vous allez devoir initialiser une instance de structure SERVICE\_TRIGGER. Cette structure n'étant pas exposée par le Framework .Net, il vous faut la définir :

```
[StructLayout(LayoutKind.Sequential)]
public struct SERVICE_TRIGGER
{
    public uint dwTriggerType;
    public uint dwAction;
    public IntPtr pTriggerSubtype;
    public uint cDataItems;
    public IntPtr pDataItems;
}
```

Il faut aussi définir tous les sous-types de triggers que vous pourriez utiliser :

```
private static readonly Guid NETWORK_MANAGER_FIRST_IP_ADDRESS_
_ARRIVAL_GUID
    = new Guid("4F27F2DE-14E2-430B-A549-7CD48CBC8245");
```

Dans un vrai projet, le recours à des énumérations serait préférable. Toutes les constantes utilisées ici sont décrites dans [http://msdn.microsoft.com/en-us/library/dd405512\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/dd405512(VS.85).aspx).

Vous pouvez alors initialiser les champs :

```
SERVICE_TRIGGER serviceTrigger = new SERVICE_TRIGGER();
// 2 pour SERVICE_TRIGGER_TYPE_IP_ADDRESS_AVAILABILITY
serviceTrigger.dwTriggerType = 2;
// 1 pour SERVICE_TRIGGER_ACTION_SERVICE_START
serviceTrigger.dwAction = 1;
```

Il faut ensuite préciser que c'est l'attribution d'une adresse IP qui est attendue :

```
IntPtr pSubTypeGuid = Marshal.AllocHGlobal(Marshal.SizeOf(typeof(Guid)));
Marshal.StructureToPtr(NETWORK_MANAGER_FIRST_IP_ADDRESS_ARRIVAL_GUID,
    pGuidIdAddressArrival, false);
serviceTrigger.pTriggerSubtype = pSubTypeGuid;
```

Enfin, il reste à appeler l'API **ChangeServiceConfig2** pour associer le démarrage du service à l'événement (voir <http://www.pinvoke.net/default.aspx/advapi32/ChangeServiceConfig2.html> pour la définition de sa signature en version .Net). Pour faire cet appel, vous avez besoin d'une nouvelle structure : SERVICE\_TRIGGER\_INFO. Cette structure permet de spécifier à quel triggers le service est associé (par exemple, la connexion et la déconnexion).

```
[StructLayout(LayoutKind.Sequential)]
public struct SERVICE_TRIGGER_INFO
{
    public uint cTriggers;
    public IntPtr pTriggers;
    public IntPtr pReserved;
}
```

L'appel se faisant alors de la manière suivante :

```
// Création du pointeur sur l'événement déclencheur
IntPtr pServiceTrigger = Marshal.AllocHGlobal(Marshal.SizeOf(typeof(
    SERVICE_TRIGGER)));
Marshal.StructureToPtr(serviceTrigger, pServiceTrigger, false);

// Création de la structure listant les événements déclencheurs (ici 1)
SERVICE_TRIGGER_INFO serviceTriggerInfo = new SERVICE_TRIGGER_INFO();
serviceTriggerInfo.cTriggers = 1;
serviceTriggerInfo.pTriggers = pServiceTrigger;

// Création du pointeur sur la structure SERVICE_TRIGGER_INFO
IntPtr pServiceTriggerInfo = Marshal.AllocHGlobal(Marshal.SizeOf(typeof(
    SERVICE_TRIGGER_INFO)));
Marshal.StructureToPtr(serviceTriggerInfo, pServiceTriggerInfo, false);

// Changement de la configuration du service
ServiceController sc = new ServiceController(serviceName);
bool bSuccess = ChangeServiceConfig2(
    sc.ServiceHandle.DangerousGetHandle(),
    8, pServiceTriggerInfo);
```

Une fois cela fait, reste à faire un peu de ménage :

```
Marshal.FreeHGlobal(pSubTypeGuid);
Marshal.FreeHGlobal(pServiceTrigger);
Marshal.FreeHGlobal(pServiceTriggerInfo);
sc.Dispose();
```

## Une dernière précaution

En théorie, votre service est maintenant configuré pour démarrer lors de l'affectation de la première adresse IP. Mais si votre service cible des systèmes Windows 7 et plus ancien, vous aurez certainement besoin de le détecter dans le démarrage de votre service. En effet, sur un système Windows 7, cet événement ne se déclenche que lorsque la condition système aura été vérifiée. Alors que sur un système plus ancien, votre service démarrera tout de suite (mode automatique) et devra donc attendre et tester lui-même la satisfaction de cette condition (par exemple, en démarrant régulièrement un thread pour tester l'état de la couche IP). La solution de facilité consiste à positionner un paramètre dans la base de registre lors de l'installation du service. Mais il est aussi possible de le savoir à l'exécution en appelant **QueryServiceConfig2** (voir <http://www.pinvoke.net/default.aspx/advapi32/QueryServiceConfig2.html> pour la signature et un exemple). Comme vous le verrez en lisant l'exemple, cet API doit être appelée deux fois : une fois pour connaître la taille des données à récupérer, et une fois pour les récupérer. Passez ensuite en paramètre **dwInfoLevel** la valeur 8 (comme précédemment) et transformez le pointeur récupéré en objet SERVICE\_TRIGGER\_INFO (via **Marshal.PtrToStructure**). Si le champ **cTriggers** a une valeur supérieure à 0, c'est que le service est démarré via un trigger.

## Conclusion

Cet article n'a fait qu'aborder la question des services déclenchés. Grâce à la possibilité de s'abonner aux événements ETW, on peut envisager des scénarios plus complexes. A titre d'exemple, un constructeur pourrait définir dans ses drivers ses propres événements, afin que les développeurs puissent en tirer parti. En attendant que de prochaines versions de Windows ne viennent compléter cette liste...

■ Patrice Manac'h



# Intégrez la biométrie à vos applications avec Windows 7

La biométrie permet d'identifier de manière unique une personne en fonction de ses caractéristiques physiques. Parmi les périphériques biométriques les plus répandus, nous trouvons les lecteurs d'empreintes digitales, aujourd'hui présents sur quasiment tous les PC portables. Cette diversité a engendré des incompatibilités entre applications et drivers, d'où la nécessité d'homogénéiser les interfaces des périphériques au moyen d'un nouveau framework présent dans Windows 7 : le Windows Biometric Framework. Dans cet article, nous allons introduire le WBF et l'utiliser dans un cas concret de signature de document par certificat.

Le WBF est constitué de plusieurs parties permettant aux éditeurs d'applications et constructeurs de périphériques de lecture d'empreintes digitales d'avoir un moyen consistant à utiliser ces périphériques d'identification dans les applications. Ses principaux composants sont :

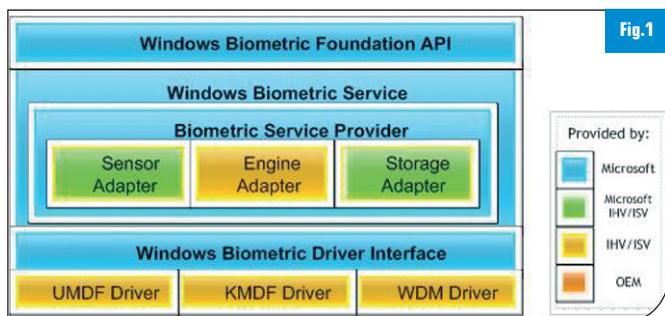
- Windows Biometric Driver Interface (WBDI) : L'interface permettant d'interconnecter un périphérique biométrique au framework.
- Windows Biometric Services (WBS): Le service qui gère les périphériques biométriques.
- WBF API: les fonctions exposées par le framework. [Fig.1]

## Utilisation dans Windows 7

Avant toute utilisation du WBF, il est nécessaire d'enregistrer une empreinte qui permettra de s'authentifier.

Pour cela, il est faut disposer d'un lecteur d'empreinte digitale et le configurer via le Panneau de Configuration (**Start > Control Panel > Biometric Devices**). [Fig.2]

Il faut ensuite enregistrer une empreinte qui permettra d'identifier l'utilisateur. [Fig.3]



Windows Biometric Framework Core Architecture.



Panneau de configuration du périphérique biométrique

Lorsqu'une empreinte est enregistrée, il devient alors possible de configurer les paramètres de biométrie pour s'authentifier sur Windows 7. [Fig.4]

## Les API du Windows Biometric Framework

Les API sont accessibles sous forme de fonctions Win32. Afin de les intégrer dans notre application .NET, nous utiliserons le wrapper et le code P/Invoke généré par PInvoker que Thomas Lebrun a mis à disposition sur son blog (voir références).

Le wrapper de Thomas Lebrun permet d'identifier un utilisateur et récupérer différentes informations telles que son nom, son SID ou encore son statut et sa description. [Fig.5]

## L'application de signature de documents

La dernière étape consiste à intégrer le Windows Biometric Framework dans une application .NET (Windows Forms ou WPF) et utiliser les informations d'identification pour récupérer un certificat permettant la signature du document sélectionné. Nous nous concentrerons uniquement sur le code relatif aux certificats et à la



Enregistrement d'une empreinte

# dossier \ windows 7

signature XML dans la suite du paragraphe. Nous laisserons libre cours à votre créativité pour implémenter le reste de l'application (choix du fichier à signer).

Le code ci-dessous récupère le chemin d'une image et instancie le wrapper du framework de biométrie :

```
private void Button_Click(object sender, RoutedEventArgs e)
{
    OpenFileDialog dialog = new OpenFileDialog();
    dialog.Filter = "Tous les fichiers (*.*)|*.*";
    dialog.FilterIndex = 0;
    if (dialog.ShowDialog() == true)
    {
        _sourceFilePath = dialog.FileName;
        fileImage.Source = new BitmapImage(new Uri(_sourceFilePath));

        _biometric = new BiometricWrapper();
        _biometric.IdentifySuccess += new BiometricWrapper.
        OnIdentifySuccessHandler(biometric_IdentifySuccess);
        _biometric.IdentifyFailed += new BiometricWrapper.
        OnIdentifyFailedHandler(biometric_IdentifyFailed);
        _biometric.IdentifyAsynchronous();
    }
}
```

Lors de la récupération de l'identifiant utilisateur par le périphérique de lecture d'empreinte digitale, nous pouvons rechercher un certificat dont le « Subject Name » correspond à l'identifiant de l'utilisateur :

```
private void biometric_IdentifySuccess(AuthenticationSuccess
EventArgs args)
{
    usernameLabel.Dispatcher.BeginInvoke(new Action(delegate
    { usernameLabel.Content = args.User; })), null);

    // Création d'un magasin de certificat utilisant le compte
    utilisateur.
    X509Store store = new X509Store(StoreLocation.CurrentUser);

    // Ouverture du magasin de certificat.
    store.Open(OpenFlags.ReadOnly);

    // Sélection d'un certificat
```

```
X509CertificateCollection certificates = store.Certificates
.Find(X509FindType.FindBySubjectName, args.User.Split('\\')
[1], true);

// Fermeture du magasin de certificat.
store.Close();

// Récupération du certificat sélectionné.
if (certificates.Count != 0)
{
    _certificate = (X509Certificate2)certificates[0];
}
}
```

Enfin la méthode finale permet de créer une signature enveloppante du fichier sélectionné. La signature électronique est générée avec la classe SignedXml qui permet d'obtenir une signature sous la forme XMLDSig, standard W3C. Le code est issu de l'article de David Grellety sur la signature électronique (voir références).

```
private void signDocButton_Click(object sender, RoutedEventArgs e)
{
    if (_certificate == null)
    {
        return;
    }

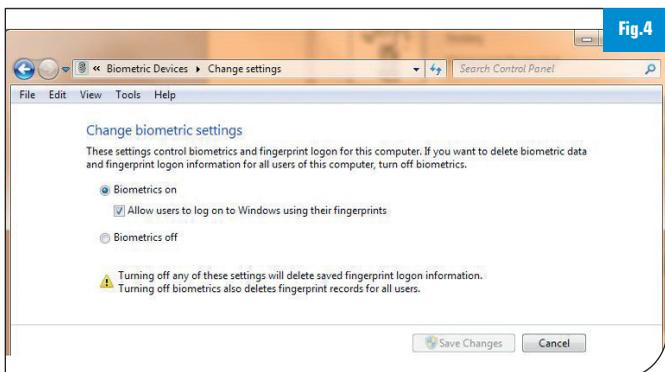
    string xmlDsigUrl = SignedXml.XmlDsigNamespaceUrl;

    // Création de l'élément XML contenant les données à signer
    XmlDocument xmlDoc = new XmlDocument();
    XmlElement element = xmlDoc.CreateElement("Content", xmlDsigUrl);
    element.InnerText = Convert.ToBase64String(File.ReadAll
Bytes(_sourceFilePath));
    xmlDoc.AppendChild(element);

    // Instanciation de l'objet SignedXml
    SignedXml signedXml = new SignedXml();

    // Définition de la clé de signature
    signedXml.SigningKey = certificate.PrivateKey;

    // Ajout des informations du certificat utilisé pour signer
    KeyInfo keyInfo = new KeyInfo();
    keyInfo.AddClause(new KeyInfoX509Data(certificate));
    signedXml.KeyInfo = keyInfo;
```



Configuration des paramètres d'authentification Windows

```
class Program
{
    static void Main(string[] args)
    {
        var biometric = new BiometricWrapper();
        biometric.IdentifySuccess += new BiometricWrapper.OnIdentifySuccessHandler(biometric_IdentifySuccess);
        biometric.IdentifyFailed += new BiometricWrapper.OnIdentifyFailedHandler(biometric_IdentifyFailed);

        biometric.IdentifyAsynchronous();

        Console.ReadLine();
    }

    static void biometric_IdentifySuccess(AuthenticationSuccessEventArgs args)
    {
        //Console.WriteLine("User that swipe on the sensor: " + args.User);
        User selectedUser = WMISeacher.FindUser(args.User);
        Console.WriteLine("User SID: " + selectedUser.SID);
        Console.WriteLine("User Name: " + selectedUser.Name);
        Console.WriteLine("User Description: " + selectedUser.Description);
        Console.WriteLine("User Status: " + selectedUser.Status);
    }

    static void biometric_IdentifyFailed(AuthenticationFailedEventArgs args)
    {
        Console.WriteLine("Identification failed: " + args.Error);
    }
}
```

Code exemple d'identification d'un utilisateur



L'application de signature de fichiers par empreinte digitale



Exemple d'un fichier signé avec XMLDSig

```
// Ajout des données à signer
System.Security.Cryptography.Xml.DataObject dataObject =
new System.Security.Cryptography.Xml.DataObject();
dataObject.Data = xmlDoc.ChildNodes;
dataObject.Id = "CONTENT";
signedXml.AddObject(dataObject);

// Création et ajout de la référence sur les données à signer
Reference reference = new Reference();
reference.Uri = "#CONTENT";
signedXml.AddReference(reference);

// Génération et sauvegarde de la signature
signedXml.ComputeSignature();
XmlElement signature = signedXml.GetXml();

string destinationFilePath = string.Format("{0}.signed
.xml", _sourceFilePath);
using (XmlTextWriter writer = new XmlTextWriter(destination
FilePath, Encoding.UTF8))
{
    signature.WriteTo(writer);
}
}
```

L'application-exemple finale se compose de 3 étapes relatives aux 3 parties de code exposées dans ce paragraphe. [Fig.6]

Le document XML final comporte le contenu du fichier et une signature vérifiable issue du certificat sélectionné grâce à l'empreinte digitale de l'utilisateur. [Fig.7]

## Glossaire

- *Biometric unit (BU)*: Une représentation commune d'un périphérique biométrique fournie par le Windows Biometric Service (WBS).
- *BU adapter*: Le composant plug-in d'un BU qui fournit un support logiciel aux fonctionnalités matérielles non supportées par un périphérique biométrique.
- *Biometric Service Provider (BSP)*: Un fournisseur de service type kernel qui s'interface avec le WBS.
- *Fingerprint Management Application (FMA)*: Une application tierce qui étend le WBF en fournissant des fonctionnalités de gestion et permet des scénarii additionnels de type Web Single Sign-On ou encore la gestion des attributs spécifiques d'un périphérique de lecture d'empreintes digitales.
- *Windows Biometric Driver Interface (WBDI)*: Une interface utilisée par les périphériques biométriques pour s'interfacer avec le WBF.
- *Windows Biometric Framework (WBF)*: Un framework disponible avec Windows 7 et qui fournit une expérience homogène et une interface commune pour tous les périphériques de lecture d'empreintes.
- *Windows Biometric Service (WBS)*: Le service qui gère tous les périphériques de lecture d'empreintes compatibles WBDI.

## Conclusion

Au travers de cette courte introduction et d'un bref exemple, nous avons montré l'utilisation simple et concrète du Windows Biometric Framework.

Cette nouveauté de Windows 7 permet de bénéficier des périphériques de lecture d'empreintes digitales directement au sein de nos applications sans se soucier des problèmes de compatibilité de drivers ou de méthodes de lecture des données du périphérique. Cette nouvelle opportunité dans le domaine de la sécurité applicative vous permettra d'innover sur de nouveaux scénarii d'authentification forte.

## Références

1. Introduction au Windows Biometric Framework : <http://download.microsoft.com/download/9/c/5/9c5b2167-8017-4bae-9fde-d599bac8184a/WBFIntro.docx>
2. Blog de Thomas Lebrun : <http://blogs.developpeur.org/tom/archive/2009/08/19/windows-seven-comment-utiliser-l-authentification-biometrique-dans-son-application-net.aspx>
3. Les API de Windows Biometric Framework : [http://msdn.microsoft.com/en-us/library/dd401509\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/dd401509(VS.85).aspx)
4. Article de David Grellety sur la signature électronique : <http://stormimon.developpez.com/dotnet/signature-electronique/>



### ■ Frédéric Queudret

MVP Client Application & CTO de la société Mpoware, accélérateur d'innovation. Société française d'édition de logiciels pour le Cloud Computing, et de prestations de services sur les technologies .NET.

<http://www.mpoware.com/>

## La migration vers Windows 7 : l'avis d'un développeur

Pour nous, la question du passage à Windows 7 était résolue d'avance : pour garder une avance technologique, nous nous devons de monter en compétence sur ce nouveau système dès sa mise à disposition. De plus, nous n'étions pas entièrement satisfaits de Vista – notamment à cause de sa lourdeur et des problèmes de compatibilité.

**P**ar ailleurs, les premiers retours de la communauté étaient très positifs : la bêta de Windows 7 semblait performante, légère et stable, avec une ergonomie retravaillée, ce qui en faisait une version très attractive.

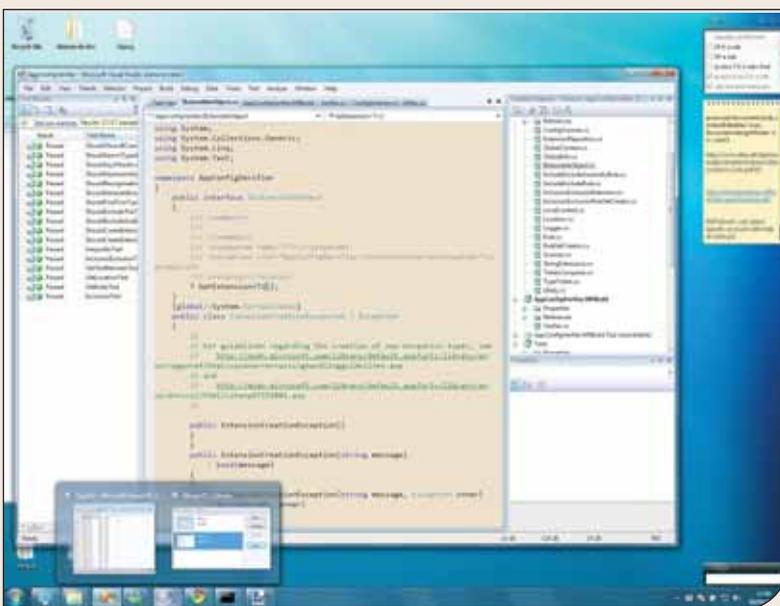
Une fois la décision prise de migrer, différentes possibilités de scénarios entrent en jeu.

Les deux plus importantes sont de migrer depuis Windows XP ou migrer depuis Windows Vista, et la différence est majeure.

Commençons par migrer depuis Windows Vista : c'est le cas le plus simple. Microsoft a annoncé une très forte compatibilité entre Vista et 7, et met à disposition une installation de mise à jour de Vista vers Windows 7 qui devrait conserver les applications et les paramètres de l'utilisateur. Cependant, il se peut que certaines applications ne fonctionnent plus, notamment celles qui font appel à la version de l'OS, ou les plug-in Internet Explorer 8 qui ne sont pas compatibles avec le système de Data Execution Prevention (DEP).

De notre propre expérience, la mise à jour fonctionne bien : nous avons passé des machines de développement complètes avec Visual Studio 2008 SP1 de Vista x64 Pro vers Windows 7 x64 Ultimate sans problème et même l'emplacement des icônes sur le bureau a été conservé.

Pour la meilleure expérience possible, il semble tout de même qu'une installation depuis zéro et réinstallation des applications avec les versions les plus à jour ou officiellement compatibles soit plus sage.

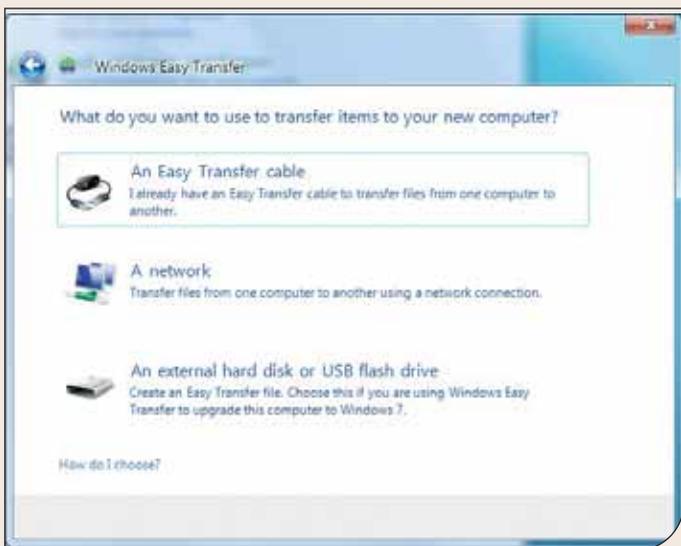


### Une histoire de migration

La migration de Windows XP vers Windows 7 est une autre histoire. En effet, Microsoft n'a pas prévu un chemin de mise à jour de Windows XP vers Windows 7, ce qui a déjà fait couler beaucoup d'encre. Le principal problème pour les entreprises qui n'ont pas encore migré vers Vista ne va pas tellement être d'évaluer Windows 7, mais de déterminer le meilleur moment pour migrer, et quel sera le temps nécessaire pour le faire. Le but étant bien d'achever cette démarche avant que le support d'XP ne prenne fin, et avant que les vendeurs d'applications ne mettent plus à jour leurs produits pour XP (ce qui devrait arriver aux alentours de 2012 d'après certains analystes). Le fait que Windows 7 partage une partie du code Windows Vista fait penser

que les problèmes de migration de XP vers Windows 7 seront équivalents aux problèmes de migration de XP vers Vista. Cependant, beaucoup de vendeurs offrent maintenant des pilotes et des mises à jour de leurs produits pour Vista, ce qui devrait permettre de passer à 7 plus facilement que lorsque Vista est sorti.

Comme évoqué précédemment, il n'y a pas de mise à jour in-place de XP à 7 : il faut donc procéder à une mise à jour personnalisée. La méthode la plus basique est de sauvegarder vos données, formater et installer le nouveau système. Ensuite, réinstaller les applications, recopier les données et reconfigurer le système. Pour simplifier ces étapes, vous pouvez utiliser l'outil gratuit Windows Easy Transfer Utility, ce qui permet de sauvegarder les préférences et les paramètres des applications ; une fois la réinstallation



terminée, vous pouvez utiliser le même utilitaire pour faire la restauration. Il y a de nombreux articles sur internet expliquant comment faire fonctionner cet outil.

Pour les entreprises, il existe des outils pour automatiser le processus comme Microsoft Deployment Toolkit 2010 (MDT) et User State Migration Toolkit 4.0 (USMT). Le MDT permet de créer des images personnalisées et de les déployer sur vos PC dans votre environnement et support Windows 7. Le USMT est l'outil équivalent de Windows Easy Transfer Utility, mais pour les entreprises. USMT et MDT peuvent être intégrés tous les deux, et permettre ainsi de réaliser automatiquement la sauvegarde, le déploiement, l'installation des applications et pilotes, et de restaurer les préférences utilisateurs. Ces outils peuvent même être utilisés pour migrer des serveurs 32bit vers 64bit.

### Analyse de Windows 7

Windows 7 démarre nettement plus vite que Vista : c'est un gain de productivité non négligeable et cela peut encourager les utilisateurs à éteindre leur PC plutôt que de le laisser en veille. La recherche semble également

assez efficace, même si Google Desktop semble avoir encore le dessus. D'un point de vue ergonomie et productivité, je trouve que la Jump List de Window 7 apporte vraiment un plus car elle permet d'avoir les documents récents par application directement accessibles via l'icône de l'application dans la barre de tâches. Le groupement par application me semble aussi intéressant, surtout pour les personnes qui ne maîtrisent pas encore trop Windows car cela permet vraiment de voir ce que l'on a ouvert.

Pour les utilisateurs qui ont la chance de posséder deux écrans avec une grande résolution, par contre, le gain de productivité qu'offre le groupement des applications n'est pas forcément évident : si par exemple on a une page web sur l'écran secondaire et que l'on clique dessus, si on veut changer de fenêtre via la barre des tâches, il faut quasiment traverser tout le bureau avec la souris, et si la fenêtre que l'on veut se trouve loin dans le groupement, on doit quasiment re-parcourir toute la distance de l'écran principal. Bref, beaucoup de chemin parcouru à la souris.

Dans le cadre d'une entreprise et

notamment pour les personnes qui occupent une place nécessitant une grande confidentialité, comme les ressources humaines ou les finances, le groupement des applications peut également poser problème : si un collègue se présente et que vous voulez lui montrer une page web par exemple en passant la souris sur l'application dans la barre des tâches, c'est toutes les fenêtres ouvertes qui sont révélées, donc si elles contiennent des informations confidentielles cela peut être gênant.

Windows 7 semble parfaitement adaptée comme machine de développement : Visual Studio 2008 SP1 fonctionne sans problème, l'OS est plus léger et les différentes fonctionnalités d'ergonomie semblent globalement payantes.



■ Frédéric Hamel  
MCTS: ASP.NET  
et WPF

Architecte et développeur passionné des technologies .NET, Frédéric Hamel travaille

actuellement à Genève dans le monde de la finance pour <http://outcoder.com>, une filiale de <http://pebbleage.ch>

PROCHAIN NUMÉRO  
N°124 - novembre 2009, parution 31 octobre

2<sup>e</sup>  
partie

**Ne manquez pas la suite de notre Dossier spécial !**

# Le guide du développeur Windows 7

Les nouveautés C++, Direct X, IE8, les API troubleshooting, 64-bit, JAC...

# GWT : comment ça marche ?

2<sup>e</sup> partie

Ce mois-ci, nous allons découvrir d'autres fonctionnalités de GWT, nous allons améliorer l'application précédente, lui permettant ainsi d'utiliser un service internet pour afficher nos photos (Picasa). Nous en profiterons aussi pour étudier la communication serveur dans GWT, les ImageBundles et l'internationalisation.



APPLICATION : WEB

LANGAGE : JAVA

SOURCE : OUI

## IMAGE BUNDLE

Dans une application, on utilise souvent beaucoup de petites icônes, pour illustrer les boutons, créer une barre d'outils, etc. Or ce qui

ralentit l'application, ce n'est pas le poids de ces images mais surtout les multiples appels au serveur pour récupérer toutes les images une par une, surtout quand certains navigateurs limitent à deux le nombre de connexions simultanées. Le principe de l'Image Bundle est de regrouper toutes les images dans une seule qui est téléchargée une seule fois et ensuite affichée en utilisant du CSS. Tout ce travail sera effectué par GWT à la compilation, avec une grande simplicité d'utilisation. [Fig.1]

Nous allons ajouter quelques images dans notre projet, un logo, une image d'espacement entre nos bouton, et deux flèches pour les boutons *suivant* et *précédent* et enfin les deux drapeaux pour changer la langue, voici l'image qui contient toutes les images :

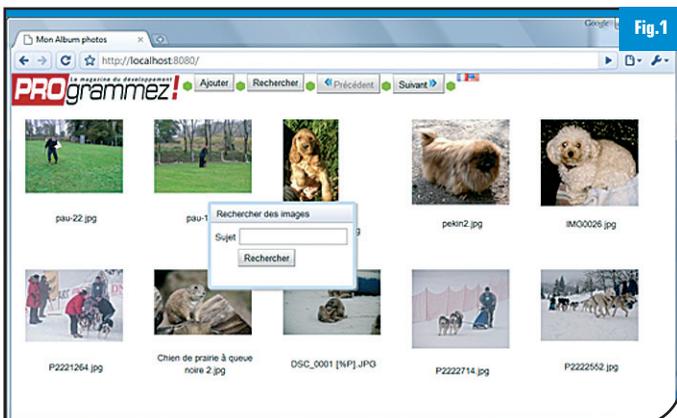


Nous avons ainsi une seule image qui contient toutes nos images. Et voici le code html automatiquement généré par GWT pour afficher le drapeau français :

```

```

On remarque qu'il affiche une image vide (clear.cache.gif) et affiche notre image en arrière-plan en la déplaçant de 250 pixels vers la gauche, 34 pixels vers le haut et en donnant la taille exacte de l'image. Au niveau programmation, la mise en place est assez simple, GWT demande la création d'une interface qui étend ImageBundle :



```
public interface MesImages extends ImageBundle {
    AbstractImagePrototype logo();
    AbstractImagePrototype precedent();
    AbstractImagePrototype suivant();
    AbstractImagePrototype spacer();
    AbstractImagePrototype fr();
    AbstractImagePrototype en();
}
```

Dans le même répertoire, il faut ajouter les images avec le même nom que la méthode mais pas forcément la même extension : *logo.jpg*, *fr.png*, *spacer.gif*. Pour utiliser ces images, il faut instancier l'interface à l'aide de la fonction `GWT.create()`, chaque méthode retourne un *AbstractImagePrototype* qui permet ensuite de créer un objet Image (avec la méthode *createImage()*), d'appliquer l'image sur un objet déjà existant (avec la méthode *applyTo()*), ou de générer le code html directement (avec la méthode *getHTML()*).

```
MesImages images = GWT.create(MesImages.class);
RootPanel.get().add(images.logo().createImage());
```

On peut de la même manière récupérer les autres images, tout est géré par GWT pour que l'utilisation soit la plus simple possible.

## INTERNATIONALISATION

Lors de la compilation, GWT optimise le code javascript généré, en particulier en réduisant la taille des fichiers téléchargés. Pour cela, il utilise la technique de "deferred binding", c'est-à-dire qu'il va générer un fichier javascript par navigateur, afin que l'utilisateur ne télécharge que le code utile pour son navigateur. Ce principe a été étendu pour l'internationalisation, chaque utilisateur ne téléchargeant que sa langue. GWT va donc créer un fichier pour chaque duo de langue/navigateur. Il va ainsi mettre directement les chaînes des caractères dans le code en fonction de la langue.

Pour gérer l'internationalisation, il utilise la technologie de Java pour écrire les fichiers de langues c'est-à-dire les fichiers propriétés. Ces fichiers sont une liste d'association clé/valeur :

```
MesConstantes.properties
ajouter = Add
rechercher = Search
```

Pour utiliser ces valeurs dans le code, il faut aussi créer une interface du même nom dans le même répertoire que le fichier de propriétés, avec les clés en nom de méthode :

```
MesConstantes.java
public interface MesMessages extends Constants {
```

```
String ajouter();
String rechercher();
}
```

Et de la même manière, on utilise `GWT.create` pour récupérer la classe :

```
MesConstantes messages = GWT.create(MesConstantes.class);
RootPanel.get.add(new Label(messages.rechercher()));
```

Pour ajouter une autre langue, il suffit d'écrire un autre fichier propriétés :

```
MesConstantes_fr.properties
ajouter = Ajouter
rechercher = Rechercher
```

Le code final sera équivalent si nous avons écrit directement ce code pour la version anglaise :

```
RootPanel.get.add(new Label("Search"));
```

Il est possible ensuite d'écrire des messages paramétrés en utilisant l'interface `Messages` :

```
MesMessages.java
interface MesMessages extends Messages {
    String hello(String name);
}
```

```
MesMessages.properties
hello = "Hello {0}"
```

```
MesMessages_fr.properties
hello = "Bonjour {0}"
```

```
MesMessages messages = GWT.create(MesMessages.class);
RootPanel.get.add(new Label(messages.hello("World")));
```

Enfin, il faut indiquer dans le fichier `module.xml`, que l'on utilise l'internationalisation, et les langues que l'on veut utiliser :

```
<inherits name="com.google.gwt.i18n.I18N"/>
<extend-property name="locale" values="fr"/>
<extend-property name="locale" values="en"/>
```

Pour changer la langue de notre application, il faut soit passer en paramètre dans l'URL, la variable locale (<http://tutoalbumphoto.appspot.com/?locale=fr>) ou mettre la balise meta suivante dans le html :

```
<meta name="gwt:property" content="locale=fr">
```



## ACCÈS AU SERVEUR

GWT seul montre rapidement ses limites, on a souvent besoin d'une partie serveur pour récupérer des données et les sauvegarder. En GWT, on peut bien sûr y accéder avec les formats traditionnels tels que le Html, le Xml ou le Json, mais GWT apporte une technologie

beaucoup plus optimisée pour partager des données avec le serveur : GWT-RPC. GWT est ainsi capable de faire passer un objet java jusqu'à l'application, et s'occupe tout seul de la sérialisation et désérialisation. Comme pour les autres fonctionnalités, GWT utilise les interfaces pour définir le service.

Reprenons notre application d'affichage de photo et ajoutons la fonctionnalité de recherche venant du service en ligne Picasa web. Pour cela, Picasa fournit un flux RSS retournant la liste des photos pour une requête donnée. L'accès à Picasa et la lecture de la recherche se fera dans la partie serveur, car il n'est pas possible pour des raisons de sécurité en javascript (donc en GWT) d'accéder à un service qui n'est pas hébergé sur le même serveur internet depuis le navigateur.

Pour faire passer les informations de chaque image vers notre application GWT, nous allons créer une classe *Photographie*, cette classe sera utilisée à la fois sur le serveur et à la fois dans la partie GWT. Elle ne contient que des propriétés accompagnées de leurs accesseurs. Attention, cette classe doit être sérialisable, c'est-à-dire que tous les objets utilisés dans la classe doivent l'être aussi (les types de bases sont sérialisables ainsi que les collections contenues dans *java.util*). Elle doit également proposer un constructeur vide pour permettre à GWT de l'utiliser.

```
public class Photographie implements Serializable {
    public String picasaUrl;
    public String imageUrl;
    public String photoMiniatureUrl;
    public String photoTitre;
    public Photographie() {
    }
    //Accesseurs
}
```

Commençons donc par créer notre interface qui contient toutes les méthodes que l'on peut appeler à distance sur notre service, ainsi que les paramètres à passer :

```
public interface PicasaService extends RemoteService {
    public List<Photographie> getPhotos(String sujet, Integer
page) throws Exception;
}
```

Écrivons maintenant le code de la partie serveur, cette classe doit se trouver dans le package serveur et non client comme tous les autres fichiers de GWT. Cette classe étend la classe *RemoteServiceServlet* et implémente notre interface *PicasaService*. Nous n'expliquerons pas plus cette partie qui ne fait que télécharger le flux XML pour le lire et construire une liste d'objets *Photographie*.

```
public class PicasaServiceImpl extends RemoteServiceServlet
implements
    PicasaService {
    private static final int PHOTOS_PAR_PAGE = 10;
    @Override
    public List<Photographie> getPhotos(String sujet, Integer
page) throws Exception{
        ArrayList<Photographie> photos = new ArrayList
<Photographie>();
```

# développement web \ r ia

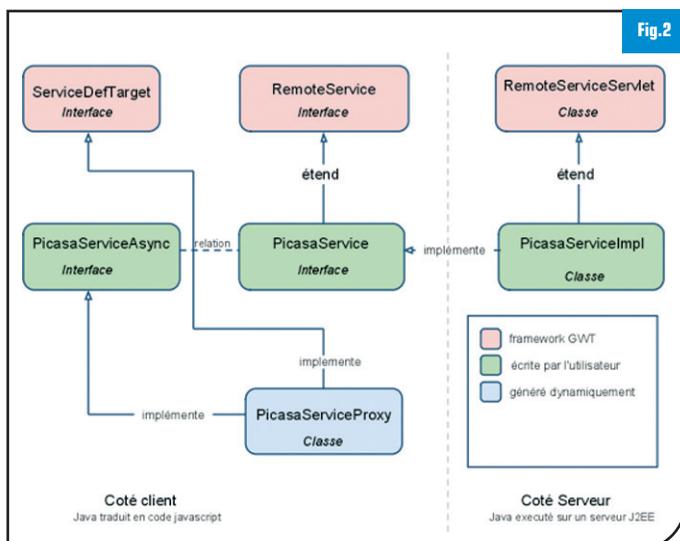
```

        DocumentBuilderFactory fabrique = DocumentBuilder
Factory
        .newInstance();
        // création d'un constructeur de documents
        DocumentBuilder constructeur = fabrique.newDocument
Builder();
        int index = 1 + page * PHOTOS_PAR_PAGE;
        URL fluxRss = new URL(
            "http://picasaweb.google.com/data/feed/api/all
?kind=photo&q="
                + sujet + "&start-index=" + index +
"&max-results="
                + PHOTOS_PAR_PAGE);
        Document d = constructeur.parse(fluxRss.openConnection()
.getInputStream());
        NodeList entrees = d.getElementsByTagName("entry");

        for (int i = 0; i < entrees.getLength(); i++) {
            Photographie pt = new Photographie(entrees.item(i)
.getChildNodes().item(9).getAttributes().get
NamedItem(
                "href").getNodeValue(), entrees
.item(i)
                .getChildNodes().item(6).getAttributes().get
NamedItem(
                "src").getNodeValue(), entrees.item(i)
                .getChildNodes().item(4).getFirstChild().get
NodeValue());
            photos.add(pt);
        }
        return photos;
    }
}

```

Maintenant que notre service est créé, il reste à l'utiliser. L'accès à un service en ligne est long, et il n'est pas possible d'attendre la réponse car il faut laisser la main au navigateur. Il faut donc utiliser un mécanisme asynchrone (le A de AJAX) avec un système de fonction de rappel.



Pour cela, il faut écrire une deuxième interface identique à l'interface du service, mais en ajoutant un paramètre qui est la fonction de retour déclenchée à la réception du résultat, et en enlevant de la signature type de retour et éventuelles exceptions lancées :

```

public interface PicasaServiceAsync {
    public void getPhotos(String sujet, Integer page, AsyncCall
back<List<Photographie>> callback);
}

```

On peut maintenant accéder à ce service de la manière habituelle, en l'instanciant avec un `GWT.create()`

```

PicasaServiceAsync picasaService = GWT.create(PicasaService
Async.class);

```

Et maintenant pour appeler le service, rien de plus simple, c'est un appel de fonction auquel on donne une fonction de retour :

```

picasaService.getPhotos(sujet, page,
    new AsyncCallback<List<Photographie>>() {
        public void onFailure(Throwable caught) {
            Window.alert("Erreur lors de l'appel de la fonction
distante getPhotos : " + caught.getMessage());
        }

        public void onSuccess(List<Photographie> result) {
            liste.clear();
            for (Photographie p : result) {
                liste.add(new Miniature(p.getPhotoMiniatureUrl(), p
                .getPhotoTitre(), p.getPicasaUrl()));
            }
        }
    });

```

Nous pouvons remarquer qu'il y a deux méthodes à implémenter pour l'interface *AsyncCallback*, la méthode *onFailure* est appelée si le serveur a renvoyé une exception (erreur sur le serveur, authentification, etc.), cela permet d'avoir facilement une gestion d'erreur. La méthode *onSuccess* retourne la valeur de retour de la fonction distante que nous avons appelée. Nous pouvons voir que le type peut être complexe, ici, c'est une liste d'objets métier que nous avons créée. Voici un schéma, qui résume les différentes classes et interfaces utilisées pour la création d'un service RPC : [Fig.2]

Il ne reste plus qu'à déployer notre code sur un serveur java tel que Tomcat ou Google App Engine. Dans un prochain article, nous verrons comment utiliser Google App Engine pour gérer les photographies directement dans la partie serveur sans utiliser le service en ligne Picasa.

Démonstration : <http://partie2.latest.tutoalbumphoto.appspot.com/>

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

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

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

■ **Adrien Ancelin et Patrice de Saint Steban**

Etudiants EFREI et EPITA

Ingénieurs d'étude et Développement, Direction Technique de SFEIR

# La sécurité dans Spring

La sécurité applicative répond à deux questions fondamentales : *l'authentification*. Qui es-tu ? Vérifier l'identité de l'utilisateur interagissant avec l'application (la méthode la plus communément utilisée étant le couple credentials username/password). Et *l'autorisation*. Qu'est ce que tu as le droit de faire ? Déterminer les habilitations d'un utilisateur authentifié au sein d'une application (les ressources auxquelles on peut accéder, les services qu'il peut appeler, les données qu'il peut consulter...)

Afin de mettre en place les mécanismes de sécurité répondant à ces deux questions, on a souvent eu recours à des solutions propriétaires, chacun gérant le problème à sa manière. Un semblant de standard est cependant apparu. A savoir le maintenant bien rôdé JAAS (Java Authentication and Authorization Service) dont la robustesse n'a d'égal que la rigidité de son implémentation. De plus, les possibilités de sécurité offertes par la spécification Servlet JEE ou la spécification EJB trahissent un manque de profondeur flagrant pour gérer des cas de scénarios d'entreprise réels. Ces différents standards ne sont d'ailleurs pas portables au niveau War ou EAR et impliquent donc un travail de reconfiguration à chaque changement d'environnement (ex : changement de serveur d'applications).

## La norme et ses limites

La norme Java Enterprise Edition permet de fournir un cadre de sécurité pour vos applications Web. Une partie de JEE (anciennement J2EE) adresse les questions de sécurité. Dans un premier temps, nous allons étudier cette norme afin d'évaluer son exhaustivité.

### Interface d'Authentification

Par déclaration dans le web.xml, il est possible de choisir entre une page JSP dédiée, un popup d'authentification etc.

### Filtrage des URL

La norme permet de réserver l'accès à certaines URI par configuration dans le web.xml. Je peux donc spécifier qu'il faut avoir le rôle ROLE\_EDITOR pour accéder à toutes les URI de type /edit/\*. Dans le code Java, la norme définit l'interface HttpServletRequest qui permet d'accéder au contexte de sécurité (voir exemple ci-dessous).

```
if (request.isUserInRole("ROLE_EDITOR")) {
    // effectue une operation en base de données
} else {
    throw new IllegalArgumentException(...);
}
```

Avec JEE 5, la plate-forme s'est enrichie de quelques annotations. Il est maintenant possible d'utiliser les annotations pour restreindre l'accès à certaines classes comme les EJB (à partir de la version 3.0) et les WebServices.

```
import javax.annotation.security.RolesAllowed;

@Stateful()
public class PanierBean implements Panier {
    @RolesAllowed("ROLE_CUSTOMER")
```

```
public void addArticle(String id) {
    // ...
}
}
```

A première vue, on pourrait donc penser que la norme fournit un support suffisant aux besoins de la plupart des projets. En creusant un peu, on se rend compte que de nombreuses fonctionnalités ne répondent pas à l'appel.

### Déclaration des utilisateurs

Dans les exemples ci-dessus, nous n'avons pas spécifié où sont stockés les utilisateurs et leurs rôles (base de données, annuaire LDAP etc.).

L'explication est simple : cet élément n'est pas standardisé par JEE. Chaque serveur d'applications doit fournir sa propre configuration pour la déclaration des utilisateurs. Si mon application doit tourner sur 5 serveurs d'applications différents, il faudra donc que je mette en place 5 configurations spécifiques.

### Limite de la portée des annotations

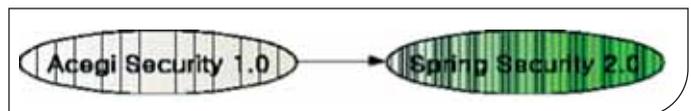
En reprenant l'exemple ci-dessus, JEE ne permet d'utiliser l'annotation de sécurité @RolesAllowed que sur un nombre limité d'artefacts : Servlets, EJB, WebServices. Cela n'inclut donc pas les contrôleurs que vous pouvez écrire avec votre framework web préféré, et aussi les DAO et les Services qui ne seraient pas écrits selon la norme EJB 3.

### Le Single Sign On

A l'heure où les applications Java sont de plus en plus nombreuses, il devient intéressant de proposer une authentification unique sur plusieurs applications. Cette fonctionnalité n'est pas prise en compte par JEE.

## Qu'est ce que Spring Security ?

Spring Security est un projet open-source spécialisé dans la sécurité Java. Ce n'est pas une extension de la norme, mais plutôt une émulation de la norme (Spring Security reste compatible avec certains éléments de syntaxe comme le request.isUserInRole(...) vu précédemment).



Acegi Security était l'œuvre de Ben Alex, développeur australien ayant une grande expérience des problématiques de sécurité. Son framework était excellent en termes de fonctionnalités mais la configuration était technique et verbeuse. Une expression avait même émergé de la blogosphère : « à chaque fois que quelqu'un utilise

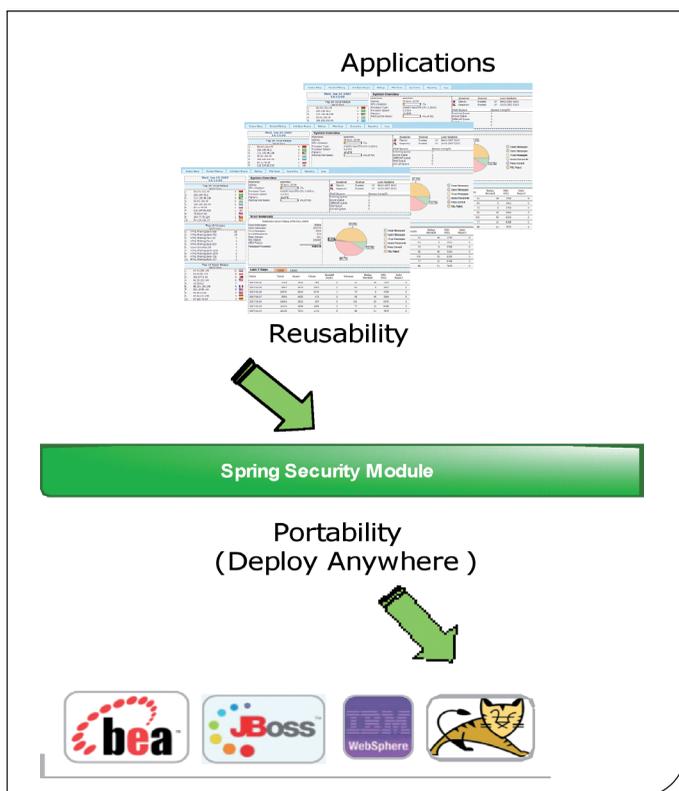
# développement web \ sécurité

*Acegi Security dans le monde, une fée s'éteint* ». Pour la version 2.0, la configuration a été complètement retravaillée. Comme vous allez le voir dans les lignes qui suivent, l'ensemble est maintenant très intuitif avec le strict minimum d'éléments techniques. Pour l'occasion, Acegi Security a été renommé en Spring Security. En terme de fonctionnalités, Spring Security offre une solution de sécurité « complète » gérant l'authentification et l'autorisation à la fois au niveau requête web (par filtrage) et au niveau invocations de méthodes/services (par programmation par aspect) voire même au niveau des objets eux-mêmes (Access Control List, ou ACL).

## Pourquoi Spring Security ?

Spring Security a pour lui de nombreux atouts pour faciliter la mise en place d'une sécurité efficace au sein de vos applications, on retiendra principalement :

- Une configuration unique et non pas spécifique pour chaque serveur d'application (la configuration a d'ailleurs été simplifiée depuis Acegi Security et est bien moins « verbeuse » que par le passé)



- Les concepts portés par Spring (injection de dépendances, programmation par aspects) pour mettre « efficacement » en place la sécurité. Développer la sécurité comme un aspect permet de s'assurer que les couches métier sont totalement indépendantes des modules d'authentification ou d'autorisation. Il est donc possible de changer totalement la politique de sécurité sans avoir à modifier les classes métier (contrôleurs, services, DAO). La sécurité est ainsi vue comme un module transverse et l'on évite ainsi la duplication de code.

- Un large support au niveau des formes d'authentification (LDAP, Single Sign On, base de données...) permet à Spring Security d'être plus souple et d'offrir des solutions de sécurité peu intrusives tout en facilitant leur intégration quelle que soit la « méthode d'authenti-

fication » côté client. Spring Security intègre même JCAPTCHA pour se protéger de toute authentification « non-humaine ».

## puSeras

*Exemple de Captcha*

- Architecture modulaire, souple et extensible permettant de développer ses propres modules d'authentification (voire de s'interfacer avec JAAS), d'autorisation...
- Intégration avec de nombreuses solutions de Single Sign On (JAAS, CAS, SiteMinder etc). En ce cas, l'identification unique permet de propager les identités au sein des applications et d'éviter les multiples logins.

## Mise en place de Spring Security

La configuration se fait dans un fichier Spring avec un namespace dédié à la sécurité.

```
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans-2.0.xsd">

<security:http access-denied-page="/accounts/denied.htm">
<security:form-login login-page="/accounts/login.htm" />
<security:intercept-url pattern="/accounts/edit*" access="ROLE_ADMIN" />
<security:intercept-url pattern="/accounts/account*" access="ROLE_ADMIN,ROLE_MEMBER" />
<security:intercept-url pattern="/accounts/**" access="IS_AUTHENTICATED_FULLY" />
<security:logout/>
</security:http>

<security:authentication-provider>
<security:password-encoder hash="md5" />
<security:user-service properties="/WEB-INF/users.properties" />
</security:authentication-provider>
</beans>
```

L'exemple ci-dessus montre la configuration de Spring Security (1). Dans cette configuration, les utilisateurs sont stockés dans les fichiers properties. Ils auraient également pu être stockés dans un annuaire LDAP ou une base de données. Par ailleurs, il est possible de sécuriser l'accès aux objets Java comme dans l'exemple ci-dessous :

```
<security:global-method-security secured-annotations="enabled">
<security:protect-pointcut expression="execution(* com.programmez
..*Service.*(..)"
access="ROLE_USER" />
</security:global-method-security>
```

D'après l'expression ci-dessus, tous les accès aux méthodes des classes finissant par « Service » seront protégés. Si vous utilisez Spring, cette expression est parsée au démarrage de l'application. Des classes « Proxy » sont générées et viennent ajouter la composante sécurité à vos classes applicatives. Tout le travail est effectué au démarrage et le surcoût à Runtime est extrêmement faible (2).

(1) Les déclarations de namespace ont été enlevées pour plus de clarté.

(2) Si vous n'utilisez pas Spring, il est possible d'utiliser AspectJ qui peut par exemple enrichir le bytecode en phase de compilation, en rajoutant les instructions de sécurité.

Si l'expression régulière ci-dessus vous paraît trop ambitieuse, il est également possible de procéder par annotations comme le montre l'exemple ci-dessous :

```
public class PanierServiceImpl implements PanierService {
    @RolesAllowed("ROLE_CUSTOMER")
    public void addArticle(String id) {
        // ...
    }
}
```

L'annotation `@RolesAllowed` avait déjà été vue dans la première partie de cet article (présentation de la norme JEE). Il s'agit bien de la même annotation, la différence étant que Spring Security permet de l'appliquer à n'importe quelle classe dans votre application (contrôleurs, Services, DAO...).

### Conclusion

Spring Security facilite grandement l'intégration de la composante sécurité avec les solutions utilisées par votre entreprise ou celles qu'elle prévoit d'employer dans le futur. Il comble également une partie des limites des spécifications Servlet JEE et EJB et offre de nouvelles perspectives en terme de sécurité, allant jusqu'à sécuriser les

objets eux-mêmes et couvrant par là-même la plupart des scénarios d'entreprises que vous serez amenés à rencontrer.

La version 3.0 de Spring Security est prévue pour la fin de l'année 2009. Elle offre de nombreuses nouvelles fonctionnalités, comme la possibilité d'utiliser des expressions avancées pour les critères d'autorisation comme dans l'exemple ci-dessous :

```
<security:http use-expressions="true">
  < security:intercept-url pattern="/secure/**"
    access="hasRole('ROLE_SUPERVISOR') and hasIpAddress('192.168.1.0/24')" />
</ security:http>
```



■ Mourad NAJAR  
Ingénieur Génie logiciel  
Centre de Services  
JAVA - STERIA



■ Ahmad SOUSAK  
Responsable cellule  
Ingénierie de Développement  
Centre de Services  
JAVA - STERIA



■ Michael Isvy  
SpringSource

## L'INFO permanente



- **L'actu** : le fil d'info quotidien de la rédaction
- **La newsletter hebdo** : abonnez-vous, comme 43 000 professionnels déjà. C'est **gratuit** !

## C'est PRATIQUE !

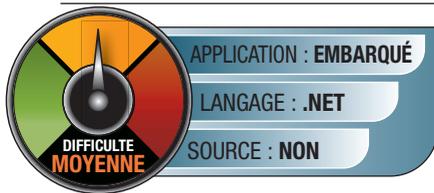
- **Le forum** : modéré par la rédaction et les auteurs de Programmez, rejoignez les forums techniques de [programmez.com](http://programmez.com)
- **Les tutoriels** : une solution en quelques clics !
- **Le téléchargement** : récupérez les nouveautés.



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

# .Net Micro Framework : l'autre visage de .Net !

Développé par l'équipe anciennement nommée SPOT (Smart Personal Object Technology), le .NET Micro Framework est une plate-forme de développement et d'exécution dédiée au monde de l'embarqué. Contrairement à ses grand-frères, le .NET Micro Framework ne nécessite pas de système d'exploitation sous-jacent pour s'exécuter.



Il embarque une couche d'abstraction, la **Hardware Abstraction Layer (HAL)** capable de discuter directement avec le matériel du périphérique sur lequel l'appli-

cation s'exécute. De ce fait, cette plate-forme est souvent qualifiée de « **bootable runtime** ». Cette couche, d'une taille d'environ 20-30 Ko fournit réellement une interface avec le hardware. En revanche, étant dépourvue de noyau, elle décharge la gestion des processus, thread et mémoire à une version allégée du **Common Language Runtime .NET (CLR)**. Au dessus de cette première couche d'abstraction, repose la **Plateforme Abstraction Layer (PAL)**. Celle-ci est responsable de la gestion des timers, des blocs mémoire, des communications asynchrones et du support de certains types .NET.

Encore au dessus, repose la **Common Language Runtime (CLR)** du .NET Micro Framework. Basée sur des standards de l'ECMA, elle est celle qui offre la possibilité d'obtenir une plate-forme de développement complètement managée : sécurité d'exécution, protection des ressources, sécurité des types, garbage *collection*, isolation des processus et threads... Elle constitue bien entendu le moteur d'exécution de la plate-forme .NET Micro Framework. Une seule application peut être déployée et lancée simultanément par la CLR du Micro Framework. C'est aussi elle qui « émule » le multithreading via une ordonnanceur de tâche coopératif.

De très petite taille, cette plate-forme embarque tout de même une **Base Class Library (BCL)**, qui bien que beaucoup plus légère que

celle présente dans le .NET Framework « classique » fournit tout de même bon nombre de bibliothèques en charge d'aider à la manipulation des flux, gestion des interfaces graphiques, des communications sur le réseau... La BCL est enrichie par l'espace de noms **Microsoft.SPOT** qui contient notamment toute la couche présentation, basée sur un modèle objet très proche de celui disponible dans **Windows Presentation Foundation** (sans XAML). [Fig.1]

Conçue pour développer des applications embarquées sur des périphériques dont la configuration matérielle est faible (processeurs ARM7 @20-40Mhz et ARM9 @200Mhz, 300Ko RAM, ~ 1Mo Flash/ROM) le processus d'exécution des applications .NET Micro Framework diffère légèrement de celui des applications .NET classiques. En effet, bien que le code C# soit compilé en **Intermediate Language (IL)**, le **Just-In-Time (JIT) compiler** n'est pas de la partie sur cette plate-forme et ce, dans un souci d'économie des ressources. Pour pouvoir s'exécuter, le code IL est compressé et tout simplement interprété par la CLR du Micro Framework. [Fig.2]

## COMMENT DÉVELOPPER ?

Le .NET Micro Framework est accessible via un SDK (Software Development Kit) téléchargeable gratuitement sur Internet. Celui-ci installe sur la machine tous les pré-requis ainsi que des exemples et de la documentation. Disponible l'an dernier en version 2.5, le .NET Micro Framework est disponible depuis Octobre 2008 en version 3.0 (donc compatible avec Visual Studio 2008).

Lors de l'installation du SDK sur un poste de développement, celui-ci intègre à partir de la version Express de Visual Studio une nouvelle

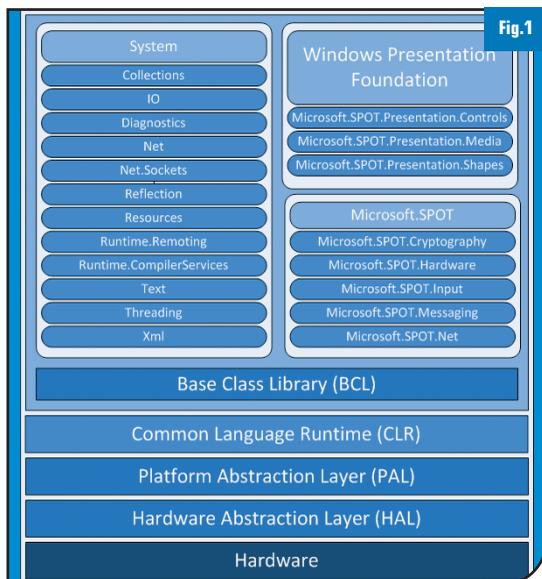


Fig.1



Fig.2

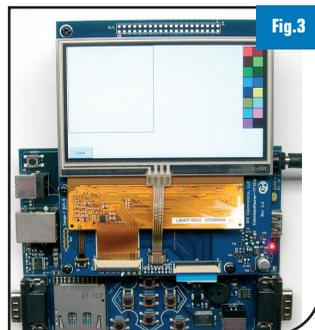


Fig.3



Fig.4

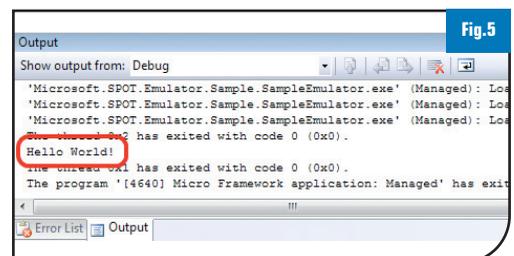


Fig.5

catégorie de modèle de projet, parmi lesquels :

- **Class Library** : Permet de créer un projet de type bibliothèque de classe pour développer et isoler la logique métier d'une application .NET Micro Framework.
- **Console Application** : Permet de créer une application console .NET Micro Framework (sans interface graphique)
- **Window Application** : Offre la possibilité de créer un projet de type application fenêtrée pour les cartes possédant un écran.

Chaque type de projet référence automatiquement la librairie « Microsoft.SPOT.Native » contenant les objets de base afin de faciliter l'écriture des premières lignes de code. Lors de la compilation et de l'exécution du projet, Visual Studio utilise pour permettre le test de l'application, un émulateur dédié au .NET Micro Framework. Celui-ci permet de simuler un déploiement sur une vraie carte à puces et de tester l'interaction de l'application avec les périphériques de base : écran LCD et boutons.

L'émulateur de base devient cependant rapidement obsolète car il n'est pas générique au point de pouvoir simuler n'importe quel hardware. Afin de démarrer tout développement en .NET Micro Framework, il est nécessaire de réfléchir au matériel ciblé par l'application : type et marque de processeur, écran ou non, réseau via Ethernet ou wifi, interaction USB, COM...). Une fois le choix effectué, il faut sélectionner une carte de développement couvrant au maximum les attentes du scénario final. Plusieurs fabricants tels que Device Solution, GHI Electronics, Digi proposent ce type de matériel, à différents niveaux et coûts (entre 150\$ et 450\$) en fonction de la sophistication du matériel. [Fig.3] et [Fig.4]

Il est possible de déployer et de debugger sur une carte de développement directement depuis Visual Studio, comme c'est le cas lors du développement sur Windows Mobile ou autre périphérique externe connecté au PC. Cependant, le déploiement pouvant être long et par souci de productivité en début de projet, chaque fabricant de carte propose également d'intégrer son propre émulateur – copie conforme de sa carte de développement - dans Visual Studio. Il est ainsi possible de choisir entre développement et émulation locale ou déploiement et débogage sur carte de développement connectée. Dans tous les cas, le switch entre émulateur et carte de développement s'effectue en quelques secondes en modifiant les propriétés du projet. Une fois le développement terminé, il est possible d'industrialiser le déploiement de l'application sur des cartes dites de production, c'est-à-dire sans périphériques additionnels inutiles. Celles-ci ont juste besoin d'être composées d'un Micro Processeur compatible .NET Micro Framework et sont électroniquement complétées avec tout le matériel nécessaire et bâties sur mesure, en fonction du programme qu'elles reçoivent : c'est à ce moment que le monde du développement .NET passe la main au monde de l'électronique.

## UNE PREMIÈRE APPLICATION

Une solution **Micro Framework Console Application**, est, du point de vue de l'environnement de développement très classique, puisque elle contient les habituels fichiers **AssemblyInfo.cs** et **Program.cs**. C'est bien évidemment ce dernier qui contient le point d'entrée de toute application, avec, comme c'est souvent le cas une méthode statique nommée « Main ».

Côté référence, la librairie **microsoft** (qui contient les types de base du Framework) est présente, accompagnée de la librairie **Microsoft.SPOT.Native**, qui, elle, apporte les types spécifiques au .NET Micro Framework. Un fichier de ressources (**Resources.resx**) est également présent, et il contient par défaut une seule entrée de

type « String » valant « Hello World ! ». L'utilisation de fichiers ressources est plus que jamais conseillée ici afin d'externaliser du code toutes les valeurs de l'application. Pour ce qui est du code généré par défaut dans le fichier « Program.cs » :

```
using System;
using Microsoft.SPOT;

namespace MFDemoProgrammez
{
    public class Program
    {
        public static void Main()
        {
            Debug.Print(
                Resources.GetString(Resources.StringResources.String1));
        }
    }
}
```

Ce code très basique a seulement pour effet d'afficher la chaîne de caractères présente dans le fichier de ressource sur la fenêtre de sortie de Visual Studio. Une pression sur la touche « F5 » du clavier suffit pour en avoir le cœur net : Le compilateur se met alors à travailler et l'émulateur prend le dessus lors de l'exécution, il s'ouvre, et se ferme immédiatement.. Pendant ce court instant, les instructions de debug ont eu le temps de s'afficher dans la console de Visual Studio. [Fig.5]

Il est, bien sûr, tout à fait possible de déboguer une application .NET Micro Framework, c'est-à-dire l'exécuter en pas-à-pas, ajouter des espions sur des variables, des points d'arrêts etc. Ceci est également possible lors de l'exécution d'une application sur une carte de développement externe. En effet, la CLR du Micro Framework embarque des outils de debug en **Remote Procedure Call (RPC)** qui rendent le débogage « distant » possible. Pour afficher le « Hello World ! » sur l'écran de l'émulateur ou d'un périphérique externe plutôt que dans la console de débogage de Visual Studio, un peu de code supplémentaire est requis. Celui-ci doit être préalablement accompagné de deux références supplémentaires pour gérer l'affichage :

- Microsoft.SPOT.TinyCore.dll
- Microsoft.SPOT.Graphics.dll

De nouveaux objets sont alors accessibles, notamment pour créer un Bitmap de la taille de l'écran et y écrire du texte à l'intérieur, via une méthode **DrawText**. Pour écrire du texte, une police (Font) est également nécessaire et doit être ajoutée en tant que ressource de l'application. En effet, les applications .NET Micro Framework étant destinées à des périphériques ayant un stockage mémoire faible, elle n'embarque que le strict minimum. Il revient donc toujours au développeur d'importer ce dont il a besoin.

Une fois une police de caractères ajoutée, une nouvelle méthode apparaît sur la classe **Resources** : **GetFont**, retournant une instance de la classe **Font**.

```
public static void Main()
{
    //récupération d'un bitmap de la taille de l'écran
    Bitmap ecran =
        new Bitmap(SystemMetrics.ScreenWidth, SystemMetrics.Screen
```

## code \ embarqué

```
Height);
//récupération du "Hello World!" dans le fichier de ressources
string texte = Resources.GetString(Resources.StringResources
.String1);
//récupération du Font
Font font = Resources.GetFont(Resources.FontResources.small);
//écriture sur l'écran
ecran.DrawText(texte, font, Color.White, 10, 10);
//appel de flush pour afficher le dessin sur le périphérique/
émulateur
ecran.Flush();
}
```

Ce code donne, à l'exécution : [Fig.6]

L'écriture de texte sur un périphérique de type écran est une opération rendue basique grâce au Framework. La gestion des boutons l'est également au travers du type d'application **Micro Framework Window Application**. Ce modèle met à disposition une application « fenêtrée » qui ne sera plus fermée directement et dans laquelle il est possible de gérer des événements tels que les pressions sur les boutons. Dans ce template de projet, Visual Studio se charge de générer une bonne partie du code requis pour bien démarrer. En effet, un fichier en particulier nommé **GPIOButtonInputProvider.cs** est d'une aide précieuse. C'est ce fichier qui est responsable du « mappage » entre le hardware et l'interface et permet donc de gérer les événements levés lorsqu'un bouton est pressé (sur l'émulateur comme sur un périphérique externe). Le fichier **Program.cs** est lui aussi bien différent de celui de l'application console de base. La class « Program » dérive ici de la classe **Microsoft.SPOT.Application** représentant une application fenêtrée .NET Micro Framework :

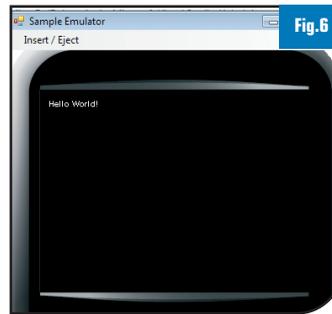
```
public class Program : Microsoft.SPOT.Application
```

Une instance de cette classe est tout d'abord créée dans le Main. Ensuite, un appel est lancé à la méthode **CreateWindow**. Cette méthode générée par Visual Studio permet de récupérer une instance de la classe **Microsoft.SPOT.Presentation.Window** représentant la fenêtre principale de l'application. C'est dans cette méthode qu'un **handler** est ajouté sur le relâchement d'un bouton. Une fois cette instance de **Windows** récupérée, un appel à la méthode **Run**

### Des changements importants pour le framework

« Récemment, nous avons migré dans la division développeur ainsi nous sommes à côté des autres technologies .Net et de l'équipe Visual Studio. Il s'agit pour nous d'un important changement dans le modèle économique. Par le passé, le Micro Framework était basé sur des royalties par unité. Avec notre rapprochement dans la division développeur, nous avons modifié notre modèle économique. Nous travaillons ainsi à mettre la plate-forme en projet partagé ou open source. Ce changement devrait lever les obstacles à son adoption sur les petits terminaux à bas prix. Cela devrait aider à faciliter son adoption dans les écoles et chez les hobbyistes. La version 4.0 (prévue pour cet automne) bénéficie de nombreuses améliorations pour faciliter l'utilisation du framework dans les applications embarquées. Tout cela est motivant pour nous. Je crois que le Micro Framework va changer profondément le développement embarqué et le simplifier. »

■ Colin Miller (équipe .Net Micro Framework)



est effectuée sur l'instance de la classe **Program**, avec en paramètre la fenêtre principale :

```
myApplication.Run(mainWindow);
```

Dans la méthode **CreateWindow**, l'appel de la méthode « **AddHandler** » nous permet d'abonner la fenêtre principale à un **RoutedEvent**, comme par exemple le relâchement d'un bouton :

```
mainWindow.AddHandler(
    Buttons.ButtonUpEvent,
    new ButtonEventHandler(OnButtonUp),
    false
);
```

Dès lors, il est possible de modifier le code de la méthode **OnButtonUp** afin de modifier l'affichage de l'élément **Text** (représenté ici par la propriété **Child** de la fenêtre principale) :

```
private void OnButtonUp(object sender, ButtonEventArgs e)
{
    string bouton = string.Empty;
    //selon le bouton pressé
    switch (e.Button)
    {
        case Microsoft.SPOT.Hardware.Button.VK_LEFT:
            bouton = "Bouton de gauche";
            break;
        case Microsoft.SPOT.Hardware.Button.VK_RIGHT:
            bouton = "Bouton de droite";
            break;
        case Microsoft.SPOT.Hardware.Button.VK_UP:
            bouton = "Bouton du haut";
            break;
        case Microsoft.SPOT.Hardware.Button.VK_DOWN:
            bouton = "Bouton du bas";
            break;
    }

    ((Text)this.mainWindow.Child).TextContent = bouton;
}
```

Le texte de l'écran est ici modifié en fonction du bouton sélectionné. [Fig.7]

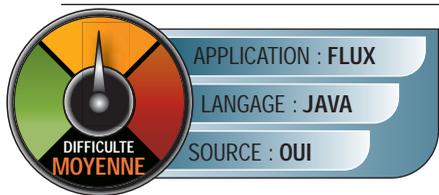
## CONCLUSION

Cette article reste une introduction au Micro Framework et présente les interactions avec le matériel proposé en standard par celui-ci. Il est bien sûr possible d'aller beaucoup plus loin avec les bibliothèques proposées par les différents constructeurs : communication Ethernet (TCP/IP, exposition et consommation de WebServices), WIFI, COM, USB, LED, Audio ou tout autre périphérique électronique connecté ! Les limites sont celles possibles en électronique et en .NET : libre à l'imagination de chacun !

■ Florent Santin (MVP – société Access it)  
et Julien Corioland (MSP – SUPINFO)

# Intégration de OSWorkflow avec les frameworks Spring IoC et Acegi

OSWorkflow propose un des choix les plus importants de systèmes de workflow pour les développeurs Java. Offrant une flexibilité importante, le framework peut s'intégrer facilement aussi bien avec un code existant, qu'avec d'autres frameworks Java. A condition d'en connaître les rouages, parce que s'il y a un point faible à OSWorkflow, c'est bien la documentation ! L'objectif de cet article est de montrer l'intégration de OSWorkflow avec Spring IoC et Acegi (Spring Security).



La configuration par défaut de OSWorkflow repose sur un fichier xml nommé `osworkflow.xml` (et qui devrait être présent sous la racine du projet ou sous le

répertoire META-INF). La structure de celui-ci devra comprendre deux points essentiels : la classe de persistance et la classe de la factory. La classe de persistance est celle qui gère la persistance des instances de workflows, des actions, des étapes...alors que la factory gère les descripteurs qui modélisent le workflow. Ci-dessous un exemple de fichier de configuration où la persistance se fait en mémoire (qui n'est bien entendu utile qu'en cas de tests) et où la factory lit (et éventuellement écrit) les modèles à partir de descriptions dans des fichiers XML.

```
<osworkflow>
  <persistence
    class="com.opensymphony.workflow.spi.memory.MemoryWorkflowStore" />
  <factory
    class="com.opensymphony.workflow.loader.XMLWorkflowFactory"
    <property key="resource" value="workflows.xml" />
  </factory>
</osworkflow>
```

En partant de ce type de configuration, le point d'entrée dont on dispose pour utiliser OSWorkflow depuis notre code, est d'instancier une classe qui implémente l'interface `com.opensymphony.workflow.Workflow` et qui, généralement, hérite de `com.opensymphony.workflow.AbstractWorkflow`. La classe fournie la plus basique (comme son nom l'indique) est `BasicWorkflow`, qu'on peut utiliser ainsi :

```
Workflow workflow = new BasicWorkflow("UtilisateurCourant");
DefaultConfiguration config = new DefaultConfiguration();
workflow.setConfiguration(config);
```

## INCONVÉNIENTS DE LA CONFIGURATION BASIQUE

Les inconvénients d'une telle utilisation sont assez clairs. En effet, l'instance du workflow qu'on a créé est associée à l'utilisateur « UtilisateurCourant ». Et comme toute application de workflow « sérieuse » fait intervenir plusieurs utilisateurs, on aura au moins une instance par utilisateur. Le deuxième inconvénient se rapporte au fait qu'il sera « lourd » de gérer un point unique et global depuis lequel on

invoque les instances du workflow pour lancer des requêtes ou effectuer des actions. C'est pour cela qu'on propose de définir un bean Spring unique dans l'application, qu'on pourra injecter dans les services qui en auront besoin et dont la récupération de l'utilisateur en cours (connecté) se fait de manière transparente à partir du contexte Acegi.

## IMPLÉMENTATION ACEGI DU CONTEXTE WORKFLOW

Pour mieux appréhender le problème d'intégration de Acegi, une compréhension du mécanisme interne de récupération de l'utilisateur (nommé "Caller" dans le vocabulaire de OSWorkflow) s'impose. En fait, la classe `AbstractWorkflow` récupère le caller à partir d'un service implémentant l'interface `com.opensymphony.workflow.WorkflowContext` en invoquant la méthode `getCaller` de celui-ci. Nous allons donc développer notre propre classe de contexte (qui implémente `com.opensymphony.workflow.WorkflowContext`) et qui renvoie le caller à partir d'un `IUserService` (un service utilisateur de base), et notre propre classe de workflow (qui hérite de `AbstractWorkflow`) qui supporte l'injection d'un contexte de façon dynamique.

```
public class IPTechWorkflow extends AbstractWorkflow {
    public IPTechWorkflow() {
    }
    public void setContext(WorkflowContext wfctx) {
        super.context = wfctx;
    }
}
```

```
public class ConnectedUserWorkflowContext implements WorkflowContext {
    private IUserService userService = null;
    public ConnectedUserWorkflowContext() {
    }
    public String getCaller() {
        return userService.getConnectedUserInfo().getUsername();
    }
    public void setRollbackOnly() {
        // Gestion de Rollback non couverte dans cet article
    }
}
```

Nous avons, dans ce code, supposé que l'interface `IUserService` dispose d'une méthode `getConnectedUserInfo()` qui renvoie une instance d'une classe "User". Un des avantages de ce type d'intégration est qu'il est complètement décorrélé du système d'authentification

## code \ java

(remarquez qu'on n'a pas encore évoqué Acegi). Il nous suffit donc, à présent, de fournir une implémentation de l'interface IUserService qui interroge le contexte Acegi pour récupérer l'utilisateur connecté :

```
public User getConnectedUserInfo() {
    Object objt = org.acegisecurity.context.SecurityContextHolder.
        getContext().getAuthentication().getPrincipal();
    return (User)objt;
}
```

### DÉFINITION DES BEANS DE CONFIGURATIONS

Définissons à présent, les beans Spring qu'on injectera :

- Le bean du service utilisateur qui récupère l'utilisateur à partir de Acegi (implémenté par une classe qu'on appellera *AcegiUserServiceImpl*)

```
<bean id="userService"
    class="com.iptech.programmez.authentication.acegi.UserServiceImpl" />
```

- Le bean du contexte workflow qui renvoie un caller à partir du service utilisateur

```
<bean id="workflow"
    class="com.iptech.programmez.workflow.IPTechWorkflow">
    <property name="context" ref="workflowContext" />
</bean>
```

- Le bean du workflow qui permettra l'injection d'un contexte

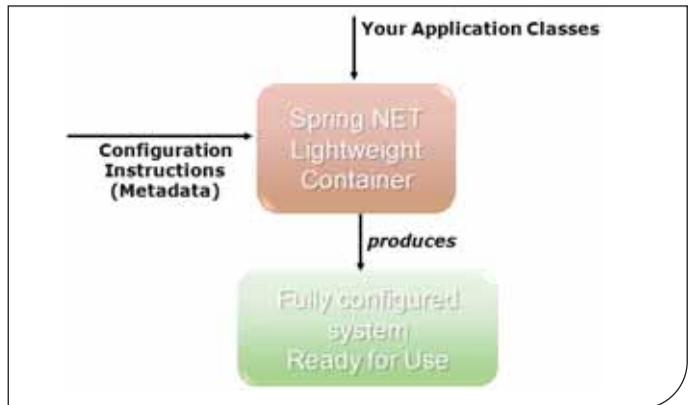
```
<bean id="workflowContext"
    class="com.iptech.programmez.workflow.ConnectedUserWorkflowContext">
    <property name="userService" ref="userService" />
</bean>
```

Nous allons, maintenant, traiter la configuration de la factory et de la persistance avec Spring. Rien ne nous empêche, bien entendu, de laisser les choses comme elles sont, et de définir cette configuration dans notre *osworkflow.xml*, mais on aurait dans ce cas deux fichiers de configuration distincts. Le but étant de définir un bean pour la factory et un bean pour la persistance et de les injecter directement dans notre bean de workflow.

La distribution de OSWorkflow vient avec la classe *com.opensymphony.workflow.config.SpringConfiguration* qui est une classe de définition de configuration, qui ne lit pas à partir de *osworkflow.xml* (comme la configuration par défaut), mais qui nous permet d'injecter directement des beans (elle expose des setteurs pour ceux-ci). L'injection de la configuration devient alors directe.

```
<bean id="workflow"
    class="com.iptech.programmez.workflow.IPTechWorkflow">
    <property name="context" ref="workflowContext" />
    <property name="configuration">
        <ref local="osworkflowConfiguration" />
    </property>
</bean>

<bean id="osworkflowConfiguration"
    class="com.opensymphony.workflow.config.SpringConfiguration">
    <property name="store">
        <ref local="idBeanDePersistence" />
    </property>
    <property name="factory">
        <ref local="idBeanDeFactory" />
    </property>
</bean>
```



### DÉCLARATION DU RESOLVER PERMETTANT L'UTILISATION DE BEANS SPRING

Nous allons à présent aborder un autre aspect de l'intégration de OSWorkflow avec Spring, et qui porte sur l'utilisation de beans Spring comme des pré-fonctions, des post-fonctions et des conditions sur les actions du workflow. En effet, lorsque nous modélisons les étapes et les actions de notre workflow, nous avons la possibilité d'indiquer au moteur de workflow qu'une condition, ou une fonction est implémentée dans un bean Spring plutôt que de lui indiquer directement le nom de la classe. Ceci se fait en injectant au bean du workflow ce qu'on appelle un « resolver ». Le resolver est utilisé par le moteur de workflow pour résoudre les types des conditions et fonctions. Dans notre cas, on s'intéresse à un resolver qui sait interpréter le type « bean spring » et qui récupère celui-ci pour l'invoquer. Un tel resolver est inclus dans la distribution de OSWorkflow, il s'agit de *com.opensymphony.workflow.util.SpringTypeResolver* qui reconnaît les conditions et les fonctions de type « spring » (en minuscule) et invoque le bean dont l'id est indiqué comme valeur de l'argument « bean.name ». On pourra donc, dans notre modélisation, utiliser le bout suivant :

```
<restrict-to>
  <conditions>
    <condition type="spring">
      <arg name="bean.name">allowOwner</arg>
    </condition>
  </conditions>
</restrict-to>
```

Et il ne nous reste que d'indiquer au moteur d'utiliser le resolver approprié :

```
<bean id="workflow"
    class="com.iptech.programmez.workflow.IPTechWorkflow">
    <property name="context" ref="workflowContext" />
    <property name="configuration">
        <ref local="osworkflowConfiguration" />
    </property>
    <property name="resolver" ref="workflowTypeResolver"></property>
</bean>

<bean id="workflowTypeResolver"
    class="com.opensymphony.workflow.util.SpringTypeResolver" />
```

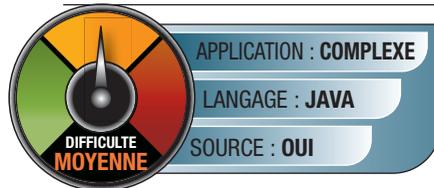
Nous arrivons à la fin de cet article où on a pu mettre en place un bean Spring de workflow qu'on pourra injecter dans nos classes de services, où la récupération de l'utilisateur connecté se fait de manière transparente à partir du contexte Acegi.

■ Hamdi Douss

Expert Java/JEE chez IP-Tech, SSII tunisienne partenaire nearshore d'éditeurs et SSII français. <http://www.iptech-offshore.com>

# À la découverte du modèle de composants **Fractal**

Le modèle de composants Fractal a été conçu à partir du début des années 2000 par France Télécom R&D et l'INRIA pour répondre aux besoins de construction, de déploiement et d'administration (par exemple : observation, contrôle, reconfiguration dynamique) de systèmes logiciels complexes.



Comme avec les modèles EJB, CCM, .NET, ou plus récemment SCA, il s'agit toujours d'obtenir des entités logicielles composables aux interfaces spécifiées

contractuellement, déployables et configurables, mais contrairement à ces modèles qui visent essentiellement la construction d'applications métiers, Fractal permet d'adresser aussi les couches inférieures comme les systèmes d'exploitation ou intergiciels (middleware), tout en fournissant une plate-forme légère et performante pour ne pas pénaliser les performances des applications.

Le modèle de composants Fractal est un ensemble de concepts permettant de décrire une architecture logicielle et de manipuler celle-ci à l'exécution, à des fins d'introspection ou de (re)configuration.

La base du développement Fractal réside dans l'écriture de composants et de liaisons permettant aux composants de communiquer. Un composant Fractal est une entité d'exécution accessible par des interfaces. Une interface est un point d'accès à un composant. Une interface implante un *type d'interface* qui spécifie les opérations supportées par l'interface. Il existe deux catégories d'interfaces : les interfaces serveurs qui correspondent aux services fournis (*provided*) par le composant, et les interfaces clients qui correspondent aux services requis (*required*) par le composant.

Un composant Fractal peut être *primitif* ou *composite*. On dit que le composant est *primitif* quand son contenu (*content*) est implémenté dans un langage de programmation, typiquement par une classe dans le monde Java, il peut alors posséder des attributs configurables. Un composant est appelé *composite* lorsque son contenu est un ensemble de sous-composants. Le modèle de composants Fractal est ainsi hiérarchique, c'est-à-dire que les sous-composants peuvent eux même contenir des sous-composants. Pour décrire ces composants *composites* on utilise un langage de description d'architecture à la XML : Fractal ADL. Celui-ci permet notamment de changer l'architecture du logiciel sans avoir à recompiler le code Java. Le modèle Fractal ne dépend pas d'un langage de programmation particulier. Néanmoins on se limitera dans cet article à l'implantation Java.

## INSTALLATION DE F4E

Si vous avez déjà Eclipse d'installé, vous pouvez installer F4E en allant dans Help > Software Updates... cliquez sur Add site... et ajoutez l'url <http://fractal.ow2.org/f4e/update-site/>. De façon alternative, vous pouvez aller à l'adresse [http://forge.ow2.org/project/showfiles.php?group\\_id=22](http://forge.ow2.org/project/showfiles.php?group_id=22) et télécharger l'archive contenant Eclipse avec F4E installé, il suffit alors de décompresser l'archive et de lancer Eclipse.

## VOTRE PREMIÈRE APPLICATION FRACTAL

Nous allons, pour cette première application, créer un petit logiciel permettant l'affichage et le calcul de prix d'un menu de restaurant. Pour les lecteurs assidus de Programmez !, l'exemple est repris de l'article sur SCA publié dans le N°110. On utilisera l'éditeur graphique de F4E pour créer le fichier de description d'architecture (ADL) du composant composite *Restaurant.fractal* ainsi que des annotations Java particulières sur le code métier pour générer les fichiers ADL des composants primitifs.

La **figure 1** montre l'architecture de l'application Fractal que nous allons construire. Le composant composite *restaurant.Restaurant* est formé de 4 sous-composants primitifs :

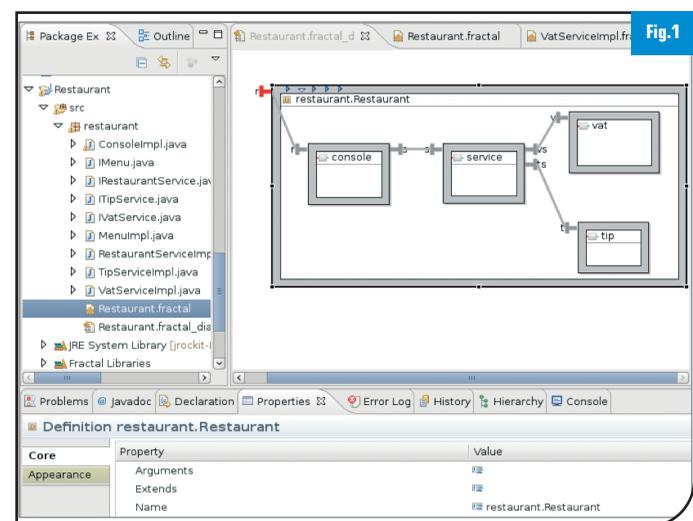
- **console** permet d'interagir avec les entrées sorties, d'afficher les menus à l'utilisateur et de saisir son choix,
- **service** donne la liste des menus et calcule l'addition pour un menu,
- **vat** calcule la TVA,
- **tip** calcule le pourboire.

Commençons à présent l'application.

Créez un nouveau projet Fractal nommé *Restaurant*.

## DÉFINITION DES INTERFACES ET DES IMPLANTATIONS

Dans le répertoire src de votre projet, créez le package *restaurant*. Nous allons dans un premier temps définir les interfaces Java qui seront associées aux interfaces Fractal de nos composants. Créez les interfaces Java suivantes :



Architecture de l'application

## code \ java

```
public interface IVatService {
    double getPriceWithVat(double price);
}
```

```
public interface ITipService {
    double getPriceWithTip(double price);
}
```

```
public interface IMenu {
    String printMenu();
}
```

```
public interface IRestaurantService {
    IMenu[] getMenus();
    double getBill(IMenu menu);
}
```

Nous allons à présent définir l'ensemble des implantations de nos composants primitifs. Pour décrire nos composants primitifs on utilise des annotations Java. Celles-ci seront ensuite utilisées pour générer les fichiers Fractal ADL et les fichiers Java des implantations modifiées de nos composants primitifs.

Le composant primitif vat est implémenté par la classe *VatServiceImpl* (annotation `@Component`). Ce composant offre une interface serveur (provides) de nom *v* et de signature *IVatService* (annotation `@Interface(name = "v", signature = restaurant.IVatService.class)`). Le composant possède aussi un attribut de nom *vatRate* initialisé à 19.6 pour représenter le taux de la TVA (annotation `@Attribute`).

```
@Component(provides = @Interface(name = "v",
    signature = restaurant.IVatService.class))
public class VatServiceImpl implements IVatService {
    @Attribute(value="19.6")
    private double vatRate;

    public double getPriceWithVat(double price) {
        return price*vatRate/100+price;
    }
}
```

```
@Component(provides = @Interface(name = "t",
    signature =restaurant.ITipService.class))
public class TipServiceImpl implements ITipService {
    @Attribute(value="10")
    private double tipRate;

    public double getPriceWithTip(double price) {
        return price*tipRate/100+price;
    }
}
```

```
public class MenuItem implements IMenu {
    private int id;
    private String details;

    MenuItem(int idC, String detailsC) {
        id = idC; details = detailsC;
    }
}
```

```
public String printMenu(){return details;}
int getId(){return id;}
}
```

Le composant *service* qui permet de calculer l'addition a besoin d'un service pour calculer la TVA et d'un autre pour calculer le pourboire. Les services requis sont représentés en Fractal grâce aux interfaces clientes et l'annotation `@Requires` permet de les définir.

Ici le composant primitif service a deux interfaces clientes, l'une de nom *vs* et de signature *IVatService*, l'autre de nom *ts* et de signature *ITipService*.

```
@Component(provides= {@Interface(name="s",
    signature=restaurant.IRestaurantService.class)})
public class RestaurantServiceImpl implements IRestaurantService {
    private IMenu[] menus;
    private double[] prices;

    @Requires(name = "vs")
    private IVatService vs;

    @Requires(name = "ts")
    private ITipService ts;

    public RestaurantServiceImpl() {
        menus=new IMenu[] {
            new MenuItem(0, "Grilled hamburger with French fries"),
            new MenuItem(1, "Roasted chicken with vegetables"),
            new MenuItem(2, "Duck breast in an orange sauce"),
            new MenuItem(3, "Duck foie gras & mango chutney")
        };
        prices = new double[] {10,15,35,50};
    }

    public double getBill(IMenu menu) {
        double menuPrice=prices[(MenuItem)menu.getId()];
        double priceWithTaxRate=vs.getPriceWithVat(menuPrice);
        double priceWithTipRate=ts.getPriceWithTip(priceWithTaxRate);
        return priceWithTipRate;
    }

    public IMenu[] getMenus(){return menus;}
}
```

Le composant *console* expose une interface de nom *r* de type *java.lang.Runnable*. Celle-ci est le point d'entrée de notre application, elle permet d'afficher le menu, d'attendre le choix du client et de retourner l'addition de son repas.

```
@Component(provides= {@Interface(name="r",signature=java.lang.
    Runnable.class)})
public class ConsoleImpl implements java.lang.Runnable {
    @Requires(name = "s")
    public IRestaurantService restaurantService;

    public void run(){
        IMenu[] menus=restaurantService.getMenus();
    }
}
```

```

System.out.println("--Menu--");
int i=0;
for(IMenu m : menus){
    System.out.println(++i + " - " + m.printMenu());
}
System.out.println("-----");

BufferedReader br =
    new BufferedReader(new InputStreamReader(System.in));

int choice=-1;
try{
    while(choice==-1){
        System.out.println("? ");
        String c = br.readLine();
        try{
            choice=new Integer(c).intValue();
            if(choice<1 || choice>menus.length){choice = -1;}
        }catch(NumberFormatException e){}
    }
}catch(IOException e){}
System.out.println();
IMenu menu=menus[choice-1];
System.out.println("Your choice: " + menu.printMenu());
System.out.println("Price: " + restaurantService.getBill(menu));
}
}

```

## DÉFINITION DE L'ASSEMBLAGE FRACTAL

Positionnez vous sur le répertoire *src*, faites un clic droit, dans le menu contextuel, un sous-menu Fractal doit être présent, allez-y et cliquez sur « Generate Fractal ADL File », cela va permettre de générer les fichiers de description des composants primitifs dans le répertoire *bin*. Pour créer de manière graphique le composant composite *Restaurant* :

- Clic droit sur le package restaurant de votre projet > New > Other...
- Sélectionnez Fractal Diagram > Next,
- Entrez *Restaurant* comme nom de fichier > Finish.

Deux fichiers sont créés : *Restaurant.fractal*, le fichier XML décrivant l'architecture Fractal et *Restaurant.fractal\_diagram*, contenant les données graphiques. L'éditeur graphique Fractal est ouvert automatiquement. Ouvrez la fenêtre de propriétés (clic droit sur le diagramme > sélectionnez Show properties view). Sélectionnez le

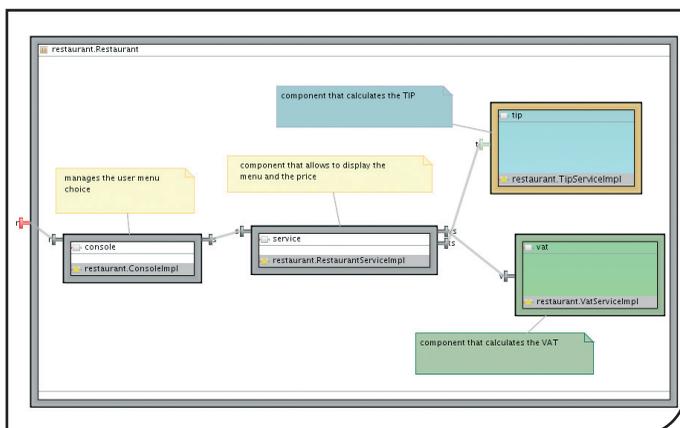


Fig.2

```

Console Problems @ javadoc Declaration
<terminated> restaurant.Restaurant [Fractal ADL] /usr/local/
--Menu--
1 - Grilled hamburger with French fries
2 - Roasted chicken with vegetables
3 - Duck breast in an orange sauce
4 - Duck fois gras & mango chutney
-----
?
1
Your choice: Grilled hamburger with French fries
Price: 13.156

```

composite et changez son nom en *restaurant.Restaurant* dans la fenêtre de propriétés.

Puis avec la palette ajoutez-lui une interface ayant pour rôle *server*, comme nom *r* et comme signature *java.lang.Runnable*. On va ensuite rajouter les 4 sous-composants implantés précédemment :

- créez le premier sous-composant, nommez le *console* et entrez *restaurant.ConsoleImpl* dans le champ *definition* dans la fenêtre de propriétés;
- créez le second sous-composant, nommez le *service* et entrez *restaurant.RestaurantServiceImpl* dans le champ *definition*;
- créez le troisième sous-composant, nommez le *vat* et entrez *restaurant.VatServiceImpl* dans le champ *definition*;
- créez le quatrième sous-composant, nommez le *tip* et entrez *restaurant.TipServiceImpl* dans le champ *definition*.

Le champ *definition* permet d'importer la description d'un composant à partir d'un autre fichier Fractal.

Faites un F5 pour rafraîchir la vue, et faire ainsi apparaître le contenu des différents composants précédemment définis.

Il faut maintenant connecter les différents composants entre eux. Pour cela, utilisez l'outil *binding* de la palette et liez :

- l'interface *r* du composant *restaurant.Restaurant* à l'interface *r* du composant *console*,
- l'interface *s* du composant *console* à l'interface *s* du composant *service*,
- l'interface *vs* du composant *service* à l'interface *v* du composant *vat*,
- l'interface *ts* du composant *service* à l'interface *t* du composant *tip*.

Maintenant que l'architecture est complètement décrite, allez sur le répertoire *src*, faites un clic droit puis lancez l'action Fractal > Generate Fractal Java File, cela génère un ensemble de fichiers Java dans le répertoire *spooned*, et permet de compiler ceux-ci dans le répertoire *bin*.

Une fois cette étape effectuée, vous pouvez démarrer l'application (figure 2). Sélectionnez le fichier *Restaurant.fractal*, faites un clic droit puis Run As > Fractal ADL.

## CONCLUSION

Cet article vous a donné un bref aperçu du modèle de composants Fractal et du plugin F4E. Vous pouvez obtenir des informations supplémentaires sur le site <http://fractal.objectweb.org/>

■ Yann Davin - Ingénieur INRIA - [yann.davin@inria.fr](mailto:yann.davin@inria.fr)

# Ecrire un module d'extension pour Drupal

Drupal, CMS très puissant que nous vous avons présenté le mois dernier est construit autour d'une architecture modulaire. Apprendre comment augmenter les fonctionnalités de Drupal par l'écriture d'un nouveau module d'extension va être notre but aujourd'hui.



Drupal est un CMS Open Source écrit en PHP. Très puissant à la base, ses possibilités n'ont guère de limites car Drupal est extensible à

volonté grâce à des modules d'extensions. Aussi surprenant que cela puisse paraître vu de l'extérieur, écrire un module d'extension pour Drupal est aisé. Les points les plus difficiles sont d'avoir une vue d'ensemble de son architecture et de son fonctionnement et d'être familiarisé avec la documentation ([www.drupal.org](http://www.drupal.org)), pas des plus claires lorsqu'on aborde la question pour la première fois. Cette étape franchie, les choses deviennent vite aisées, ainsi que cet article se propose de vous le démontrer par la pratique. Peu de pré-requis sont en outre nécessaires: des notions de base de PHP et avoir déjà manipulé un peu Drupal en tant qu'administrateur.

## 1 L'ORGANISATION DE DRUPAL

Drupal s'articule autour de concept très forts: les nœuds, les thèmes, les hooks. Tout contenu (avec quelques exceptions toute-fois) est un nœud, stocké en base de données. Les thèmes s'occupent de l'apparence des pages rendues par Drupal. Les thèmes sont un sujet d'article à part entière, peut être pour une autre fois. Et enfin les hooks. Ils effectuent les traitements et ils constituent l'essentiel de la logique entre données/nœuds et rendu/thèmes. Fondamentalement Drupal est un tout petit runtime, en grande partie lui-même constitué de modules ;) et dont le rôle est de gérer l'installation des modules, et de recenser les hooks que ces modules contiennent pour les invoquer au moment opportun. Vu ainsi, Drupal présente de fortes analogies avec Eclipse (ou même avec les hooks d'Emacs ;). Avec Eclipse (cf Programmez! 63) on donne des fichiers XML pour décrire un plugin et des classes Java implémentant des interfaces bien définies. Avec Drupal on donne un fichier .info pour décrire le module et on implémente des fonctions respectant une convention de nommage bien précise. Ces fonctions sont les hooks. Même si Drupal, pour des raisons historiques n'est pas orienté objet, le lecteur familier des plugins Eclipse ne sera pas dépaycé. Et pour le lecteur totalement étranger à Eclipse, c'est facile aussi. Nous allons ainsi voir ensemble comment construire un module de type bloc, c'est-à-dire un module qui affiche un contenu dans une zone d'une page. Puis nous verrons comment faire collaborer ce module avec son environnement, c'est-à-dire son intégration dans l'interface d'administration de Drupal, les dépendances éventuelles avec d'autres modules, les permissions d'accès et nous allons même ajouter une fonctionnalité de configuration du module.

## 2 LES OUTILS

Il ne faut rien d'autre qu'un éditeur de texte pour réaliser ce travail. Toutefois, il est bon de s'appuyer aussi sur des modules de développement que vous trouverez sur le site de Drupal. Je vous recom-

mande tout particulièrement les modules Coder et Devel. Le premier vérifiera la bonne forme de votre code et vous donnera des conseils pour le rendre plus "Drupalish" le cas échéant. Le second module, Devel propose un grand nombre de fonctionnalités. Nous en utiliserons quelques-unes ensemble, le lecteur découvrira seul les autres. S'il fallait n'en utiliser que deux ce seraient probablement les fonctionnalités de réinstallation des modules et de vidage du cache. Il est donc pertinent, quoique pas obligatoire, d'installer ces modules avant de poursuivre la lecture de cet article

## 3 DÉFINIR LE MODULE PAR UN FICHIER .INFO

Un module d'extension doit être "déclaré" à Drupal. Pour cela, on doit d'abord créer un répertoire du nom du module, nous choisissons ici *programmez* comme nom, sous l'arborescence sites/all/modules de Drupal. Ensuite, on doit y déposer un fichier *programmez.info*. Cette convention de nommage est à respecter. Ce fichier est un fichier 'PHP INI'. Voici le contenu du nôtre pour notre très simple module :

```
;$Id$
name = "Programmez"
description = "Demo de programmation Drupal"
core = 6.x
php = 5.1
```

Le signe cabalistique de la première ligne est un emplacement pour des informations CVS. En effet Drupal et ses modules sont maintenus par un server CVS. Et comme cet emplacement CVS n'est pas une directive du fichier .info, il est placé en commentaire par le point-virgule qui le précède ;)

Le nom du module, sur la ligne suivante sera affiché par Drupal dans les pages d'administration. Donc on donnera un nom propre sur lui :) La chaîne de description apparaît elle aussi dans l'interface d'administration de Drupal et permet d'afficher des informations complémentaires au sujet du module. Enfin les directives *core* et *php* spécifient les versions minimales de Drupal et PHP pour que le module fonctionne correctement.

## 4 LE FICHIER DE CODE DU MODULE

Il devra être déposé à côté du fichier *programmez.info* et il devra être nommé *programmez.module*. Comme cette extension ne l'indique pas, il contiendra du code PHP, avec une particularité. On y mettra la balise ouvrante `<?php` mais on n'y mettra pas la balise fermante `?>` C'est une convention Drupal. Son rôle est notamment d'éviter la présence de caractères blancs dans les sorties du script. Le code doit aussi comporter un emplacement CVS, comme dans le fichier .info Un code Drupal propre sur lui, "Drupalish", doit obéir à un certain nombre d'autres conventions que vous trouverez à cette page : <http://drupal.org/coding-standards>. Et le module *Coder* vérifiera à

votre demande si vous respectez ces conventions. Ce respect n'est pas obligatoire mais vivement conseillé. Nous arrivons donc au code de notre module, code qui contient des commentaires également conventionnels.

```
<?php
// $Id$

/**
 * @file
 * Module Programmez
 *
 */

/**
 * Implementation de hook_block
 */

/*
function programmez_block($op='list' , $delta=0, $edit=array()) {
  switch ($op) {
    case 'list':
      $blocks[0]['info'] = t('Programmez!');
      return $blocks;
    case 'view':
      $blocks['subject'] = t('Demo Programmez!');
      $blocks['content'] = t('Abonnez vous ! :-)');
      return $blocks;
  }
}
```

Une seule petite fonction et notre module est opérationnel comme on le voit [Fig.1]. Cette fonction mérite toute notre attention car c'est un hook. Pour commencer, cette fonction n'est pas nommée n'importe comment. Son modèle, si l'on peut dire, est *hook\_block*. On pourra consulter sa documentation en faisant une recherche sur ce nom dans les pages des API sur le site de Drupal. Le fait que notre fonction contienne le terme block assure que Drupal affichera la valeur retournée dans un bloc. Ensuite, comme notre fonction doit agir dans le contexte de notre module, son nom contient celui du module. Tout ceci implique que notre fonction ne peut être nommée que *programmez\_block*. Cette fonction sera invoquée par le runtime de Drupal à

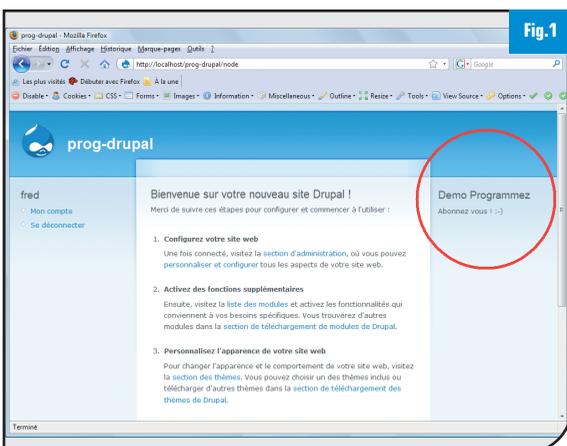
des moments différents. Lorsqu'il s'agit d'afficher le contenu du bloc bien évidemment, mais dans l'interface d'administration des blocs également. Le contexte d'invocation est déduit de l'argument *\$op* passé au hook par le runtime. Ici la valeur *list* correspond au contexte d'administration et *view* à l'affichage. Il existe d'autres valeurs. Nous renvoyons le lecteur à la documentation. Une dernière remarque: toutes les chaînes sont transformées par la fonction *t*. Ceci n'est pas obligatoire, mais très vivement conseillé. Par l'appel à *t()* Drupal vérifie si le langage préféré par l'utilisateur est différent du langage par défaut. Si oui et si un fichier de traduction est disponible pour cet autre langage, les textes seront traduits automatiquement.

## 5 ACTIVER LE MODULE ET L'AFFICHAGE DU BLOC

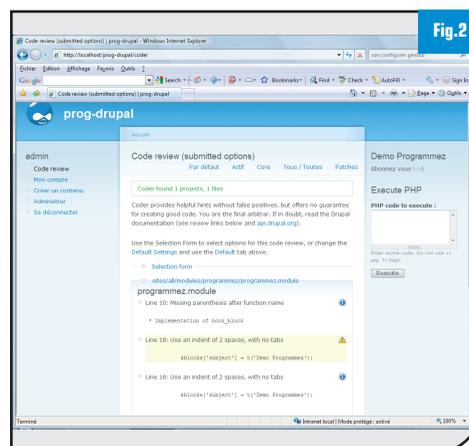
Pour cela, il suffit de se rendre dans la page d'administration des modules et de cocher la case devant le module *Programmez*. Ensuite, on se rend dans la page d'administration des blocs et on affecte le bloc de notre module à une zone de l'affichage. Dans l'illustration [Fig.1] nous avons choisi la colonne de droite.

## 6 UTILISER LES MODULES DE DÉVELOPPEMENT

Notre code est-il parfaitement écrit ? Est-il drupalish ? Pour le savoir, et si vous avez installé le module *Coder*, allez dans le menu navigation en colonne de gauche. En tête de menu vous trouverez une entrée "Code review". Sélectionnez-là. Vous arrivez à une page. Demandez par exemple une revue de code avec tous les détails (minor(most)) sur un seul module, le module *programmez* qui doit apparaître dans la liste. Validez. Drupal examine votre code et vous retourne ses remarques, comme illustré [Fig.2]. Nous voyons que Drupal critique l'indentation de notre code (code disponible sur notre site). Une fonctionnalité à ne pas manquer des modules de développement est le 'themer'. Logé en administrateur vous verrez dans un coin des pages un petit rectangle avec une case à cocher. Cliquez dans la case puis immédiatement après cliquez sur une zone d'une page. La zone de rendu du bloc Programmez est un bon choix :) Vous verrez alors s'ouvrir une fenêtre popup [Fig.3] qui vous renseignera sur le cheminement suivi par Drupal pour aboutir à l'affichage du bloc, et vous pourrez aussi consulter la valeur de quelques variables. Enfin le module *Devel* vous permet d'afficher un bloc dans lequel vous pouvez saisir du code PHP pour interprétation immédiate [Fig.4]. Ne manquez pas d'en user et abuser.



Notre module de démonstration en action.



Drupal trouve à redire sur l'indentation de notre code.



La fonctionnalité themer vous permet de comprendre le rendu d'un bloc.

## 7 DÉFINIR DES DÉPENDANCES ENTRE MODULES

Notre module est si simple qu'il ne dépend d'aucun autre. Ce n'est pas toujours le cas. Les dépendances se définissent dans le fichier `.info` du module. Supposons, pour l'exemple, que notre module dépende des modules `node` et `user`, on écrira

```
dependencies[] = node
dependencies[] = user
```

Cela suffit pour que ces dépendances soient prises en compte dans l'interface d'administration des modules, comme on le voit illustré [Fig.5].

## 8 IMPLÉMENTER L'AIDE CONTEXTUELLE

Vous l'avez sans doute deviné, ceci se fait au moyen d'un hook qui se nomme `hook_help`. Suivant les principes expliqués pour notre premier hook, nous pouvons écrire notre fonction :

```
/**
 * Implementation de hook_help
 */

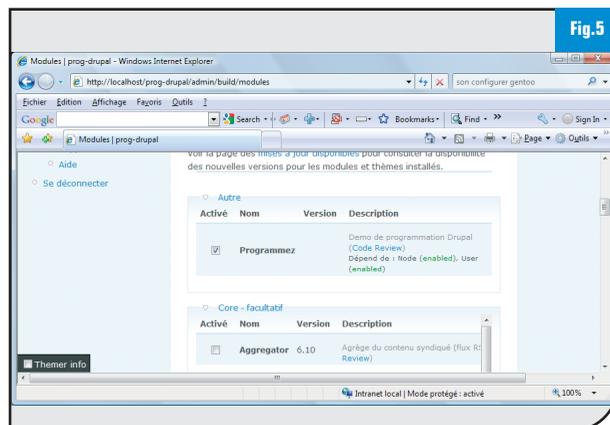
function programmez_help($path, $arg) {
  switch ($path) {
    case 'admin/help#programmez':
      return '<p>'. t('Programmez! vous aide sur le forum de son site') . '</p>';

    case 'admin/build/block':
      return t('<p>Pour bien Programmez! le mieux est de vous abonner ;)</p>');
  }
}
```

C'est tout simple. Selon le contexte Drupal invoque notre hook en lui passant en argument une URL correspondant à ce contexte. Le lecteur consultera la documentation en ce qui concerne la constitution de ces URL. Ici nous traitons l'URL de la page d'administration des blocs, `admin/build/block` et la capture ci-dessous nous montre l'aide apparaissant dans cette page [Fig.6]. Et nous traitons également l'URL `admin/help#programmez` qui correspond à l'aide relative à notre module dans les pages d'aide générales, comme le montre la capture [Fig.7].



Le module de Devel vous offre un interpréteur PHP interactif.



Nous avons ajouté des dépendances à notre module

## 9 DÉFINIR DES PERMISSIONS

Tous les modules bien conçus permettent de définir des permissions afin de limiter l'accès aux contenus. Dans l'univers Drupal, il n'est pas rare qu'un contenu ne soit accessible qu'à l'administrateur ou à des utilisateurs enregistrés, mais invisible pour des visiteurs anonymes. Nous voulons faire la même chose pour notre bloc Programmez. Cela est beaucoup plus simple qu'il n'y paraît. D'abord un hook, `hook_perm`, invoqué lors de la construction de la page 'Administrer!Gestion des utilisateurs!Droits d'accès' permet de définir les champs dotés de cases à cocher qui seront affichés dans cette page pour notre module :

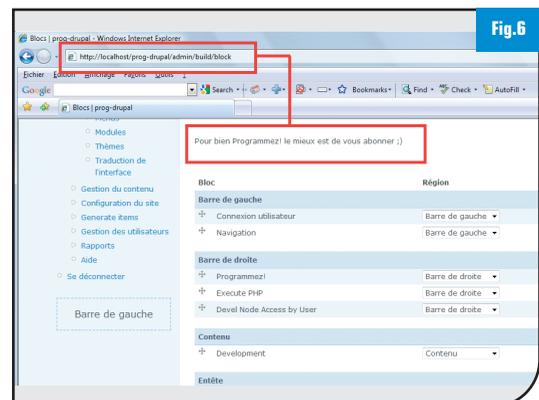
```
/**
 * Implementation de hook_perm
 */

function programmez_perm() {
  return array(t('acces contenu Programmez!'),
    t('administration Programmez!'));
}
```

Moyennant quoi, nous pouvons dire dans l'interface d'administration que seuls les utilisateurs authentifiés ont le droit de voir notre bloc [Fig.8]. Ensuite et enfin, nous modifions notre tout premier hook comme ceci :

```
function programmez_block($op='list' , $delta=0, $edit=array()) {
  switch ($op) {
    case 'list':
      $blocks[0]['info'] = t('Programmez!');
      return $blocks;
    case 'view':
      if(user_access('acces contenu Programmez!', $user))
      {
        $blocks['subject'] = t('Demo Programmez!');
        $blocks['content'] = t('Abonnez vous ! :-)');
      }
      return $blocks;
  }
}
```

Tout se tient dans l'usage de la fonction `user_access` que nous invoquons avec deux arguments. D'abord une chaîne rigoureusement identique à celle donnée plus haut pour l'interface d'administration,



Notre aide contextuelle apparaissant dans la page d'administration des blocs.

ensuite l'utilisateur courant qui est dans la variable globale \$user. Si l'utilisateur est authentifié nous retournons le contenu du bloc, sinon nous retournons un bloc vide.

## 10 DÉFINIR UNE CONFIGURATION DU MODULE

Un module complexe peut avoir besoin d'être configuré. On présentera alors un formulaire de configuration dans l'interface d'administration. Les formulaires HTML c'est toujours pénible à manipuler ! Mais fort heureusement, Drupal propose des fonctionnalités très évoluées pour les formulaires. Dans notre exemple nous créons un formulaire contenant un seul champ d'édition, pour la saisie d'un texte additionnel. Vous l'aviez deviné, ceci se fait au moyen d'un hook :

```
/**
 * Implementation de hook_admin
 */
function programmez_admin() {
  $form['programmez_supplement'] = array(
    '#type' => 'textfield',
    '#title' => t('texte supplémentaire'),
    '#default_value' => variable_get('programmez_supplement', ''),
    '#size' => 20,
    '#maxlength' => 20,
    '#description' => t("Donner un texte supplémentaire a afficher dans le bloc"),
    '#required' => FALSE,
  );
  return system_settings_form($form);
}
```

Dans ce code #required positionné à FALSE permet de laisser le champ de saisie vide. Ensuite vient la question de savoir comment accéder à ce formulaire ? Un bon moyen est d'ajouter une entrée dans le menu d'administration pour y accéder. Nous avons choisi le menu de configuration du site (chemin 'admin/settings/programmez') et l'ajout de l'entrée se fait au moyen d'un hook, cela ne surprendra plus personne :)

```
/**
 * Implementation de hook_menu
 */
function programmez_menu() {
```

```
$items = array();

$items['admin/settings/programmez'] = array(
  'title' => 'Configuration Programmez!',
  'description' => 'La configuration permet de donner un texte supplémentaire a afficher',
  'page callback' => 'drupal_get_form',
  'page arguments' => array('programmez_admin'),
  'access arguments' => array('access administration pages'),
  'type' => MENU_NORMAL_ITEM);

return $items;
}
```

Le point fort de ce code est la déclaration de la fonction de rappel (page\_callback) que Drupal exécutera lorsque l'administrateur cliquera sur l'entrée de menu. Il s'agit ici de *drupal\_get\_form* qui va construire notre formulaire. Notez qu'en argument nous lui passons le hook qui définit le formulaire. Tout est maintenant en place. Nous adaptions une dernière fois notre premier hook afin d'ajouter l'affichage du texte supplémentaire.

```
function programmez_block($op='list', $delta=0, $edit=array()) {
  switch ($op) {
    case 'list':
      $blocks[0]['info'] = t('Programmez!');
      return $blocks;
    case 'view':
      if(user_access('acces contenu Programmez!', $user))
      {
        $blocks['subject'] = t('Demo Programmez!');
        $blocks['content'] = t('Abonnez vous ! :-)');
        $blocks['content'] = $blocks['content'] . '<p>' . variable_get('programmez_supplement', '') . '</p>';
      }
      return $blocks;
    }
  }
}
```

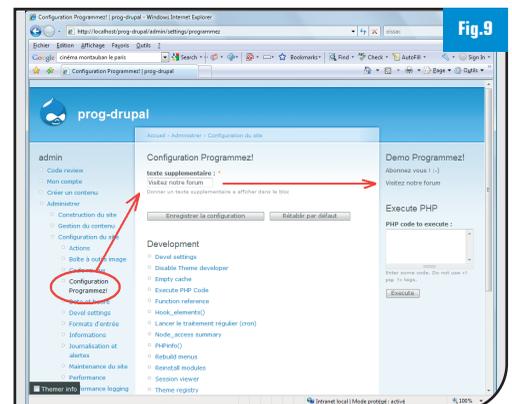
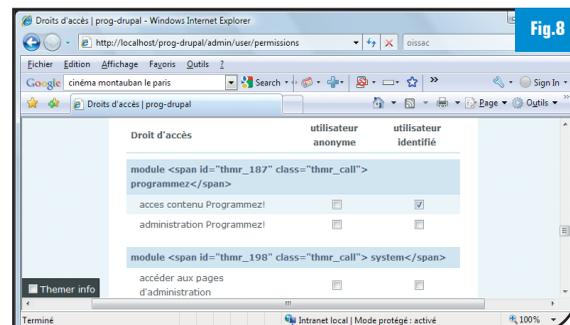
Le résultat des courses est illustré par notre dernière capture [Fig.9]. Nous disposons désormais d'un module complet. Avec bien peu de travail finalement. Vive Drupal !

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



▲ Définition des droits d'accès au contenu de notre module.

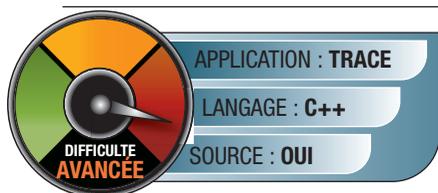
◀ Le texte d'aide de notre module dans les pages d'aide en ligne de Drupal.



Notre module est maintenant doté d'une fonctionnalité de configuration

# Tracez et hackez les processus sous Linux

Imposer un point d'arrêt à un processus, examiner le contenu de sa mémoire, modifier le comportement d'un appel, sont autant de choses que vous pouvez réaliser avec une seule API: *ptrace*.



Linux c'est le paradis du programmeur passionné, du hacker. Ce système met tellement d'outils à notre disposition que l'on n'est jamais au bout des découvertes. Ainsi saviez-vous que votre Linux contenait un utilitaire du nom de *strace* ? *Strace* est un outil de débogage qui permet de surveiller les appels systèmes effectués par un programme ainsi que les signaux qu'il reçoit. Vous avez un programme qui ne se comporte pas comme attendu mais vous ne savez pas pourquoi ? Alors vous pouvez le lancer via *strace* et découvrir, par exemple, que le dit programme essaie de lire un fichier inexistant et que c'est ce qui le fait avorter. *Strace* fournit un volume d'informations incroyable. Faisons un essai en lançant la commande *ls* avec lui :

```
strace ls
```

Comme l'illustre la capture [Fig.1], *strace* émet une sortie très abondante. Voici les toutes premières lignes émises par *strace* :

```
execve("/bin/ls", ["ls"], [/* 38 vars */]) = 0
brk(0) = 0x8704000
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
mmap2(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb7f86000
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
```

Il est intéressant de remarquer en tout premier la présence d'un appel système, *execve*, dont la présence, si l'on considère la commande *ls* seulement, peut surprendre. Nous allons bientôt comprendre la raison de cette présence. Ensuite nous voyons l'appel système *brk*, qui concerne la gestion de la mémoire. Puis viennent

trois appels à la fonction système *access*, dont le rôle est de vérifier si un processus a la permission d'accéder à un fichier. Nous voyons que ces trois appels émettent chacun une erreur. Mais cela n'empêche pas *ls* de fonctionner pour autant. Si vous examinez la sortie entière de *strace ls*, vous verrez le cheminement que fait votre système Linux pour charger une librairie. Vous pouvez, si vous voulez, pour examiner les choses à tête reposée, rediriger la sortie de *strace* dans un fichier en prenant bien soin de considérer qu'une grande partie des messages sont des messages d'erreur, donc émis sur *stderr*. Vous procéderez donc ainsi :

```
strace ls >strace-ls.txt 2>&1
```

pour avoir la totalité de la sortie. Tout cela est déjà fort intéressant en soi, mais en tant que programmeur nous nous interrogeons: *ls* est sans doute compilé sans information de débogage, alors comment l'utilitaire *strace* procède-t-il pour obtenir un tel résultat ? La réponse est: avec une seule API, mais une API magique, la fonction *ptrace*, que nous apprenons maintenant à utiliser. Comme *ptrace* fonctionne à un très bas niveau, ce que nous dirons sera lié à une architecture, le i386 en l'occurrence, et devra être adapté à une autre architecture le cas échéant.

## 1 APPELS SYSTÈMES

Fondamentalement, *ptrace* agit en amont et en aval des appels systèmes. Qu'est-ce que les appels systèmes ? Ce sont des passages en mode noyau via l'interruption 0x80 du microprocesseur, les registres EAX, EBX, ECX, EDX, ESI et EDI étant chargés avant l'interruption. Après celle-ci le résultat de l'appel est dans le registre EAX. Le programmeur Linux ne manipule normalement jamais cela directement. C'est la *glibc* qui s'en charge pour lui. Mais pour utiliser *ptrace* avec profit, nous devons savoir comment les choses se passent. Nous allons nous intéresser à la fonction système *write* et l'invoquer dans un Hello World! en assembleur (*helloworld-as.s* sur

```
fred@mars:~$ strace ls
execve("/bin/ls", ["ls"], [/* 37 vars */]) = 0
brk(0) = 0x8712000
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
mmap2(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb802e000
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
open("/tls/i686/sse2/cmov/librt.so.1", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/tls/i686/sse2/librt.so.1", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/tls/i686/cmov/librt.so.1", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/tls/i686/librt.so.1", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/tls/sse2/cmov/librt.so.1", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/tls/sse2/librt.so.1", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/tls/cmov/librt.so.1", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/tls/librt.so.1", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/i686/sse2/cmov/librt.so.1", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/i686/sse2/librt.so.1", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/i686/cmov/librt.so.1", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/i686/librt.so.1", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/sse2/cmov/librt.so.1", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/sse2/librt.so.1", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/cmov/librt.so.1", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/librt.so.1", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/home/fred/lib/tls/i686/sse2/cmov/librt.so.1", O_RDONLY) = -1 ENOENT (No such file or directory)
stat64("/home/fred/lib/tls/i686/sse2/cmov", 0xbf44bbdc) = -1 ENOENT (No such file or directory)
open("/home/fred/lib/tls/i686/sse2/librt.so.1", O_RDONLY) = -1 ENOENT (No such file or directory)
stat64("/home/fred/lib/tls/i686/sse2", 0xbf44bbdc) = -1 ENOENT (No such file or directory)
open("/home/fred/lib/tls/i686/cmov/librt.so.1", O_RDONLY) = -1 ENOENT (No such file or directory)
```

Application de *strace* sur la commande *ls*.

```
fred@mars:~/Developpement/ptrace
WRITE(2)
Programmer's Manual
WRITE(2)

NAME
write - write to a file descriptor

SYNOPSIS
#include <unistd.h>

ssize_t write(int fd, const void *buf, size_t count);

DESCRIPTION
write() writes up to count bytes from the buffer pointed buf to the file referred to by the file descriptor fd.

The number of bytes written may be less than count if, for example, there is insufficient space on the underlying physical medium, or the RLIMIT_FSIZE resource limit is encountered (see setrlimit(2)), or the call was interrupted by a signal handler after having written less than count bytes. (See also pipe(7).)

For a seekable file (i.e., one to which lseek(2) may be applied, for exam
Manual page write(2) line 1
```

La page de man de la fonction *write*

notre site). Pour cela la question est: comment charger les registres ? La convention Linux est tout ce qu'il y a de simple: EAX est chargé avec le numéro de l'appel système. Ce numéro se trouve dans le fichier `/usr/include/asm/unistd_32.h`. Nous y apprenons que `write` est affublé du numéro 4. Pour les autres registres, on consulte la page de *man* qui nous apprend [Fig.2] que la fonction `write` reçoit en arguments un descripteur de fichier, un pointeur sur les octets à écrire et le nombre d'octets à écrire. Et bien les registres sont chargés dans cet ordre, tout simplement. Le principe est toujours le même, et si cet article s'intéresse principalement à `write`, vous n'aurez aucun mal à torturer l'appel système de votre choix de la même façon. Voici notre code assembleur :

```
.section .data
hello:
    .ascii "Hello World!\n"
.section .text
.globl _start
_start:
    movl $4, %eax
    movl $1, %ebx
    movl $hello, %ecx
    movl $13, %edx
    int $0x80
    movl $1, %eax
    movl $0, %ebx
    int $0x80
```

Pour obtenir un exécutable, procédez avec ces deux commandes

```
as -o helloworld-as.o helloworld-as.s
ld -o helloworld helloworld.o
```

On remarque à la fin du code un appel à la fonction système `exit`. Tout programme Linux propre sur lui doit se terminer ainsi. Le numéro de l'appel pour `exit` est 1 et on passe la valeur zéro en paramètre pour indiquer le succès des opérations.

## 2 PÈRE, FILS ET SIGNAUX

Lorsqu'un programme est manipulé par `ptrace`, il émet des informations au moyen de signaux Linux. Les signaux sont des informations qui circulent entre processus. Nous devons donc avoir deux processus. Classiquement en programmation Linux, un processus père et son fils. Non moins classiquement le père se dédouble avec un appel à `fork`, puis se met à l'écoute de son fils. Le processus dédoublé, le fils donc, invoque `ptrace` sur lui-même, donc demande à être tracé, mais immédiatement après, il appelle `execl` pour substituer à lui-même notre programme `helloworld-as` qui sera donc celui qui enverra les signaux au père. Si vous n'êtes pas familier avec cette cuisine, il vous suffit d'appliquer la recette. Nous étudierons processus et signaux Linux en détail une autre fois. Voici donc un premier code (`demo1.cpp` sur le site) qui ne se comporte pas très bien :

```
#include <iostream>
using namespace std;

#include <sys/ptrace.h>
#include <sys/types.h>
#include <sys/wait.h>
```

```
#include <unistd.h>

void ecoute_fils()
{
    int status;
    wait(&status);
    cout << "Un signal a émané du fils" << endl;
    // pour éviter un mauvais comportement sur la console
    sleep(1);
}

int main(int argc, char* argv[])
{
    pid_t fils;

    switch(fils = fork())
    {
        case -1:
            cout << "Erreur, impossible de forker" << endl;
            break;
        case 0:
            ptrace(PTRACE_TRACEME, 0, NULL, NULL);
            execl("./helloworld-as", "./helloworld-as", NULL, NULL);
            break;
        default:
            ecoute_fils();
            break;
    }
    return 0;
}
```

Le problème est que le fils s'arrête au premier appel système qu'il rencontre, et finalement son père meurt avant lui. Nous devons affiner le code de la fonction `ecoute_fils` en ce qui concerne la lecture des signaux et demander au fils de reprendre son cours après un premier arrêt. Notre fonction devient (cf. `demo2.cpp` sur le site pour le code complet) :

```
void ecoute_fils(pid_t fils)
{
    int status;
    for(;;)
    {
        wait(&status);
        if(WIFEXITED(status)
        {
            cout << "Processus fils terminé normalement" << endl;
            break;
        }
        if(WIFSIGNALED(status))
        {
            cout << "Processus fils terminé à cause d'un signal" << endl;
            break;
        }
        cout << "Un signal a émané du fils" << endl;
        ptrace(PTRACE_CONT, fils, NULL, NULL);
    }
}
```

## code \ linux

Deux macros permettent de mieux savoir dans quelle condition le fils s'est terminé ou si le signal provient de *ptrace*. Un appel à *ptrace* avec `PTRACE_CONT` en argument permet au fils de continuer. Le comportement est maintenant correct. `Demo3.cpp` sur le site est une variante. Le seul changement par rapport à ce qui précède est :

```
ptrace(PTRACE_CONT, fils, NULL, SIGKILL);
```

Cet appel demande la continuation du fils tout en lui envoyant un signal. Ici `SIGKILL` qui tue le fils. Information qui, en tant que signal, remonte ensuite au père émetteur :

### 3 OBTENIR DES INFORMATIONS DE PTRACE

Tout cela est fort bien, mais nous ne savons toujours pas vraiment ce qui a arrêté notre processus. La rencontre d'un appel système certes, mais lequel ?

Pour obtenir le numéro de l'appel système concerné nous invoquons maintenant *ptrace* avec l'argument `PTRACE_PEEKUSER` qui peut extraire un à un les contenus des registres processus ainsi que le contenu de `EAX` avant l'appel en mémoire.

L'accès à la valeur du registre choisi se fait avec un index défini dans l'en-tête `sys/reg.h`. Attention ! Nous adressons des entiers 32 bits, ce qui requiert de multiplier par 4 l'index. Voici le nouveau code (`demo4.cpp` sur le site) de notre fonction *ecoute\_fils*

```
// pour la définition des index de registre
#include <sys/reg.h>

void ecoute_fils(pid_t fils)
{
    int status;
    long reg_eax;

    for(;;)
    {
        wait(&status);
        if(WIFEXITED(status))
        {
            cout << "Processus fils terminé normalement" << endl;
            break;
        }
        if(WIFSIGNALED(status))
        {
            cout << "Processus fils terminé à cause d'un signal" << endl;
        }
        cout << "Un signal a émané du fils" << endl;
        reg_eax = ptrace(PTRACE_PEEKUSER, fils, ORIG_EAX * 4, NULL);
        cout << "Il s'agit de l'appel système n° " << reg_eax << endl;
        ptrace(PTRACE_CONT, fils, NULL, NULL);
    }
}
```

Nous avons encore une surprise. Nous espérions l'appel système 4 mais le programme nous dit qu'il s'agit du 11, qui est *execve* invoqué par la *glibc* suite à l'invocation de *execl*. Toujours pas de *write* en vue. Avant de résoudre ce problème, mentionnons que nous pouvons, grâce à l'argument `PTRACE_POKEUSER` de *ptrace*, modifier

les valeurs contenues dans les registres, valeurs qui seront par la suite utilisées si nous continuons la trace du programme.

### 4 TRACER UN À UN LES APPELS

Si l'argument `PTRACE_CONT` relance l'exécution jusqu'à sa fin du processus en cours de traçage, l'argument `PTRACE_SYSCALL` le relance seulement jusqu'à la rencontre d'un autre appel système. C'est donc le moyen de tracer pas à pas une application, le "pas" étant l'intervalle indéterminé entre deux appels système. Pour obtenir ce résultat nous ne modifions qu'une seule ligne de l'exemple précédent :

```
ptrace(PTRACE_SYSCALL, fils, NULL, NULL);
```

Nous traçons bien maintenant le programme `helloworld-as`, mais nous rencontrons une nouvelle surprise comme en témoigne la sortie émise :

```
Un signal a émané du fils
Il s'agit de l'appel système n° 11
Un signal a émané du fils
Il s'agit de l'appel système n° 4
Hello World!
Un signal a émané du fils
Il s'agit de l'appel système n° 4
Un signal a émané du fils
Il s'agit de l'appel système n° 1
Processus fils terminé normalement
```

On dirait que l'appel à *write* (numéro 4) a été doublé. La réalité est qu'avec `PTRACE_SYSCALL` le traçage s'arrête à l'entrée ET à la sortie des appels système. Et si dans l'exemple précédent les appels 11 et 1, *execve* et *exit* respectivement, n'apparaissent qu'une fois, c'est tout simplement parce qu'ils sont des cas particulier: ils ne rendent pas la main à l'appelant ! :-)

Nous pouvons maintenant améliorer notre traçage en affichant le contenu des registres avant et après chaque appel systèmes. Vous trouverez le code complet, `demo6.cpp` sur notre site.

Ce code montre encore une nouvelle possibilité de *ptrace*. Nous avons dit plus haut que l'argument `PTRACE_PEEKUSER` permet d'obtenir, un à un, le contenu des registres. Notre code continue d'utiliser cette possibilité pour afficher le contenu d'`EAX` avant et après l'appel de la fonction. Cependant, travailler avec chaque registre l'un après l'autre est fastidieux, c'est pourquoi notre fonction *print\_regs*, qui affiche d'un coup tous les registres, utilise un nouvel argument: `PTRACE_GETREGS`. Voici maintenant un extrait de la sortie de `demo6`, concernant l'appel à *write* :

```
Entrée dans l'appel système: 4
```

```
Contenu des registres:
EAX: ffffffffda
EBX: 00000001
ECX: 08049098
EDX: 0000000d
ESI: 00000000
EDI: 00000000
EBP: 00000000
XDS: 0000007b
```

# DÉVELOPPEZ VOTRE SAVOIR-FAIRE



Langage et code, développement web,  
carrières et métier :  
Programmez !, c'est votre outil  
de veille technologique.

Pour votre développement personnel et professionnel,  
abonnez-vous à Programmez !

## Choisissez votre formule

- **Abonnement 1 an au magazine : 49 €**  
(au lieu de 65,45 € tarif au numéro) *Tarif France métropolitaine*
- **Abonnement Intégral : 1 an au magazine + archives  
sur Internet et PDF : 59 €** *Tarif France métropolitaine*
- **Abonnement PDF / 1 an : 30 €** - *Tarif unique*  
Inscription et paiement **exclusivement en ligne**  
[www.programmez.com](http://www.programmez.com)
- **Abonnement Etudiant : 1 an au magazine : 39 €**  
(au lieu de 65,45 € tarif au numéro) *Offre France métropolitaine*

11 numéros par an : 49 €\*

**Economisez 16,45 €\***

\*Tarif France métropolitaine

## + Abonnement INTÉGRAL

**ACCÈS ILLIMITÉ** aux ARCHIVES du MAGAZINE pour 0,84€ par mois !

Cette option est réservée aux abonnés pour 1 an au magazine,  
quel que soit le type d'abonnement (Standard, Numérique, Etudiant).  
Le prix de leur abonnement normal est majoré de 10 € (prix identique

pour toutes zones géographiques). Pendant la durée de leur abonnement,  
ils ont ainsi accès, en supplément, à tous les anciens numéros et articles/  
dossiers parus.

**OUI, je m'abonne** Vous pouvez vous abonner en ligne et trouver tous les tarifs [www.programmez.com](http://www.programmez.com)

### PROGRAMMEZ

- Abonnement 1 an au magazine : 49 €** (au lieu de 65,45 € tarif au numéro) *Tarif France métropolitaine*
- Abonnement Intégral : 1 an au magazine + archives sur Internet et PDF : 59 €** *Tarif France métropolitaine*
- Abonnement Etudiant : 1 an au magazine : 39 €** (joindre copie carte étudiant) *Offre France métropolitaine*

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

Nom : ..... Prénom : .....

Adresse : .....

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

Tél : ..... E-mail : .....

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

A remplir et retourner sous enveloppe affranchie à :  
Programmez ! - Service Abonnements - 22 rue René Boulanger - 75472 Paris Cedex 10.

[abonnements.programmez@groupe-gli.com](mailto:abonnements.programmez@groupe-gli.com)

**PRO** *Le magazine du développement* **grammez!**

**Offre limitée,**  
valable jusqu'au  
31 octobre 2009

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

Conformément à la loi Informatique et  
Libertés du 05/01/78, vous disposez  
d'un droit d'accès et de rectification  
aux données vous concernant.

Par notre intermédiaire, vous pouvez  
être amené à recevoir des propositions  
d'autres sociétés ou associations.  
Si vous ne le souhaitez pas, il vous  
suffit de nous écrire en nous précisant  
toutes vos coordonnées.

```

XES: 0000007b
XFS: 00000000
XGS: 00000000
EIP: 0804808a
XCS: 00000073
EFLAGS: 00010292
ESP: bfd74b30
XSS: 0000007b

```

```
Hello World!
```

```
-----
```

```
Sortie de l'appel système: 4
```

```
Contenu des registres:
```

```

EAX: 0000000d
EBX: 00000001
ECX: 08049098
EDX: 0000000d
ESI: 00000000
EDI: 00000000
EBP: 00000000
XDS: 0000007b
XES: 0000007b
XFS: 00000000
XGS: 00000000
EIP: 0804808a
XCS: 00000073
EFLAGS: 00010292
ESP: bfd74b30
XSS: 0000007b

```

```
Valeur de retour de l'appel: 13
```

Le contenu de registre nous permet de bien suivre le déroulement des opérations. Ainsi nous voyons que EDX contient bien la valeur hexadécimale 0x0D soit 13 caractères à écrire, puis à la sortie de l'appel, c'est EAX qui contient cette valeur, ce qui prouve que les 13 caractères ont bien été écrits. Et puisque nous savons maintenant arrêter la trace avant un appel système et que nous savons modifier le contenu d'un registre (ou de tous avec l'argument `PTTRACE_SETREGS`) avant l'exécution proprement dite de l'appel, nous avons le moyen d'hacker un programme :

## 5 HACKER UN PROGRAMME

Soit un programme C++ basique qui écrit "Hello World!" dans un fichier :

```

#include <iostream>
#include <fstream>

using namespace std;

int main(int argc, char* argv[])
{
    fstream fichier("helloworld.txt", ios::out|ios::binary);

```

```

fichier << "Hello World!\n";
fichier.close();
return 0;
}

```

Nous voulons hacker ce programme pour qu'il écrive "Programmez!" dans le fichier, au lieu de "Hello World!" Nous ne savons certes pas du tout comment l'écriture dans le flux est implémentée par le runtime C++, mais ce qui est certain c'est que l'implémentation finira par invoquer l'appel système `write` et c'est là que nous agirons. Quand la trace s'arrête avant l'appel système, ECX contient, forcément, le pointeur sur la zone mémoire contenant les caractères à écrire. Il nous suffit d'écrire les nôtres à la place, grâce à l'argument `PTTRACE_POKEDATA` de `ptrace` et l'affaire est dans le sac, si nous faisons bien attention au fait que l'architecture i386 est en Little Endian :) Voici, extrait de `demo7.cpp` (disponible sur notre site), le code qui réalise l'opération:

```

void poke_programmez(pid_t fils)
{
    cout << "POKE PROGRAMMEZ!" << endl;

    long reg_ecx;

    // au stade où nous sommes, les données de la chaîne
    // que le fils doit normalement imprimer sont
    // pointées par le registre ECX
    reg_ecx = ptrace(PTTRACE_PEEKUSER, fils, ECX * 4, NULL);
    // Attention au little endian :)
    ptrace(PTTRACE_POKEDATA, fils,
           reg_ecx, 0x676F7250);
    reg_ecx += 4;
    ptrace(PTTRACE_POKEDATA, fils,
           reg_ecx, 0x6D6D6172);
    reg_ecx += 4;
    ptrace(PTTRACE_POKEDATA, fils,
           reg_ecx, 0x0A217A65);
}

```

Cela fonctionne très bien. Le lecteur attentif aura remarqué que nous n'avons pas agi sur le nombre de caractères à écrire. Pour une bidouille de grande classe nous devrions :) Mais alors le runtime C++ verra que le nombre de caractères écrits diffère du nombre demandé et relancera l'appel. Pour contrer cela, il faut encore améliorer notre code, ce que nous laissons à la sagacité du lecteur.

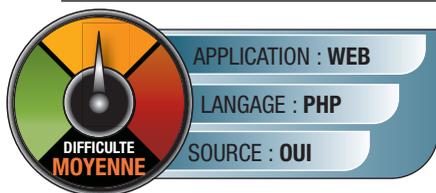
## 6 CONCLUSION

Le dernier exemple prouve que `ptrace` est très puissant, et même trop... `ptrace` est critiqué parce que sa puissance permet de programmer des rootkits aux effets dévastateurs. Il est malgré tout possible de sécuriser une application critique en la protégeant de `ptrace`. Voici deux clés: `ptrace` ne pouvant être associé qu'à un seul processus, si celui-ci se trace lui-même, il se protège. Ensuite `ptrace` utilise les signaux et le noyau Linux propose des possibilités pour capturer ceux-ci. Dans l'avenir `ptrace` est appelé à être remplacé par `D-Trace`, l'implémentation pour Linux d'une fonctionnalité très évoluée d'OpenSolaris. Mais les travaux n'en sont qu'à leurs débuts.

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

# Intégrer une webcam en Flash - PHP côté serveur

Dans la création des sites web, on distingue deux mondes assez hermétiques. D'un côté, celui des sites Flash : il s'agit de sites riches graphiquement, pourvus d'animations, mais généralement sans interaction ou presque avec une base de données, et donc sans développement côté serveur. Les sites institutionnels de marques de luxe, ou les sites de lancement de produits s'inscrivent dans cette catégorie.



De l'autre côté, le monde des sites transactionnels : il s'agit de sites plus sobres, mais qui interagissent fortement avec les bases de données et avec par conséquent beaucoup de code côté serveur. Les sites de ventes en ligne constituent bien sûr un très bon exemple. Ces deux mondes font appel à des compétences distinctes qui travaillent parfois de manière juxtaposée, mais rarement ensemble. Ces compétences n'ont pas la même formation à la base, ni le même statut : dans le premier monde on retrouve des créatifs ou graphistes, et dans le second des informaticiens.

Nous nous proposons ici d'ouvrir une porte entre ces deux mondes en partant d'un cas précis nécessitant d'utiliser Flash pour alimenter des données présentes sur un serveur. ActionScript dans sa version 3.0 est orienté objet et un développeur Java ou PHP objet ne se sentira pas dépaysé.

Adobe essaye d'ailleurs de convaincre les développeurs de migrer vers sa plate-forme Flex basée sur Flash, mais ceci dépasse le cadre de cet article.

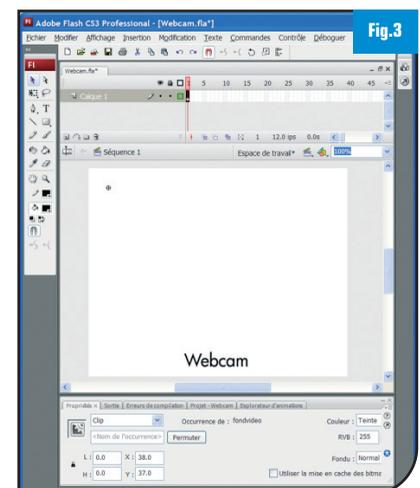
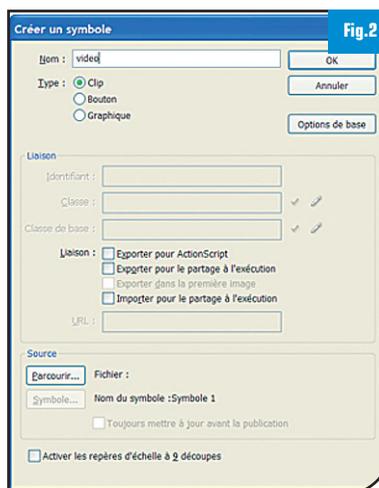
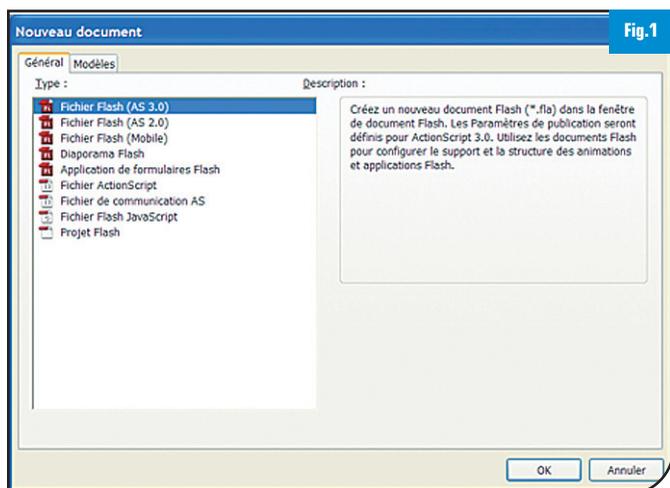
Nous souhaitons dans notre exemple permettre à l'utilisateur de fournir une photographie à intégrer dans un profil au sein d'une base

de données. Dans un développement web classique, on demanderait à l'utilisateur de réaliser un téléchargement d'une photo présente sous forme d'un fichier sur son disque dur. Mais ceci suppose qu'il utilise son ordinateur personnel et qu'il dispose bien d'une photo adéquate, ce qui peut être un frein. Nous allons donc tenter d'utiliser la webcam connectée à l'ordinateur si elle existe. La norme HTML ne supporte pas nativement les webcam, mais par contre Flash sait le faire. Nous allons donc intégrer une application Flash au sein d'une page pour prendre une photo de l'internaute ; ensuite nous verrons comment remonter cette photo vers le serveur.

## GESTION DE LA WEBCAM DANS FLASH

Nous utilisons dans cet exemple Adobe Flash CS 3.0, avec la version 3.0 d'ActionScript.

Cet exemple fonctionnera aussi avec la version CS 4.0, mais contraindra l'utilisateur à disposer au minimum de la version 9.0 du plugin Flash Player. Vous pouvez télécharger une version d'évaluation de l'outil de développement d'Adobe sur le site <http://www.adobe.fr>.



## code \ flash

Dans l'environnement d'Adobe, nous créons d'abord un nouveau projet webcam.flp auquel nous ajoutons un fichier Flash (AS 3.0) webcam. [Fig.1]

La fenêtre de propriétés nous permet de régler la taille de l'animation Flash. A l'aide de l'outil Texte, on rajoute un texte "Webcam" en bas de l'animation. On ouvre ensuite la Bibliothèque (Fenêtre > Bibliothèque) et par clic droit on insère un nouveau Symbole de type Clip que nous appelons *fondvideo*. [Fig.2]

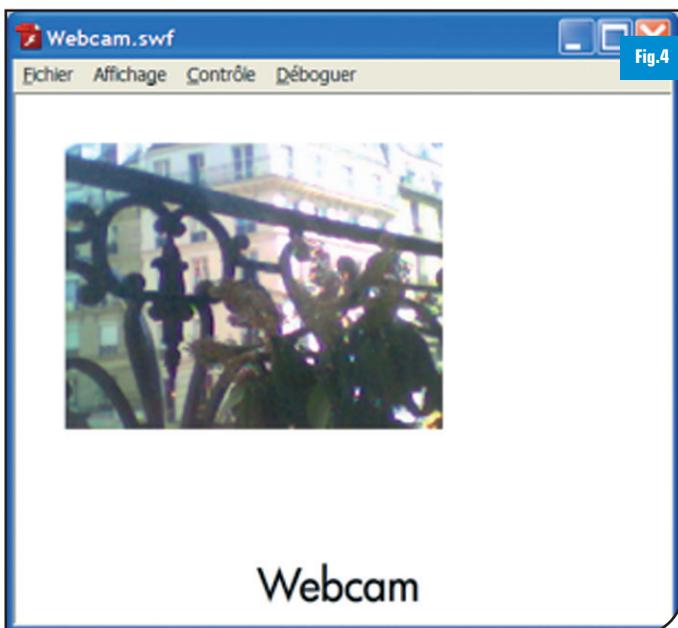
On revient sur notre Séquence1 dans la vue principale, et par glisser-déposer on y ajoute une occurrence de ce clip qu'on appelle également *fondvideo* (propriété Nom de l'occurrence). [Fig.3]

Notre clip est alors matérialisé uniquement par un petit rond qui représente son coin supérieur gauche et que nous pouvons positionner dans notre séquence. Attention : il ne faut surtout pas lui associer de couleur, car celle-ci supplanterait l'affichage de la vidéo. L'association du clip avec l'image de la webcam est réalisée par programmation. On ouvre la fenêtre de codage via le menu Fenêtre > Action, et on double-clique sur notre Séquence 1 pour associer du code au lancement de cette première séquence.

```
var cam:Camera;
var lavelideo:Video;

// On récupère la Webcam
cam = Camera.getCamera();
// Résolution de la caméra
// en fonction du modèle de webcam
cam.setMode(480,360,30,true);

// On crée une vidéo
lavelideo=new Video (480,360);
// On redimensionne suivant les dimensions
// de l'animation
lavelideo.scaleX = 0.6;
lavelideo.scaleY = 0.6;
```



```
// On associe la caméra à la vidéo
lavelideo.attachCamera(cam);
// Et la vidéo au clip de l'animation
fondvideo.addChild(lavelideo);

stop();
```

On peut alors exécuter notre projet pour voir si cela fonctionne. Vous devez autoriser l'affichage de la webcam au moment de l'exécution pour des raisons évidentes de sécurité et l'image de la webcam doit ensuite apparaître dans l'animation flash. [Fig.4]

Si ce n'est pas le cas, regardez les erreurs dans la console d'exécution et assurez-vous que votre webcam est en état de fonctionnement et que vous avez bien nommé le clip *fondvideo* dans l'animation.

Vous pouvez tester le fonctionnement au sein du navigateur web grâce au menu Fichier > Publier qui génère le fichier flash compilé et une page HTML pour contenir l'animation.

## REMONTÉE D'UNE IMAGE VERS LE SERVEUR

Maintenant que nous affichons l'image de la webcam, nous allons nous en servir pour prendre une photo et la remonter vers le serveur. Pour ce faire, nous allons mettre en oeuvre les étapes suivantes :

1. ajout d'un bouton dans l'animation et gestion de l'événement correspondant pour prendre une photo avec la webcam ;
2. encodage de la photo prise ;
3. remontée de la photo vers le serveur ;
4. prise en charge de la photo par un script PHP et enregistrement sur le disque du serveur.

### Etape 1 : prise d'une photo

Dans la bibliothèque de notre projet Flash, nous insérons un nouveau symbole de type bouton et nous le dessinons par exemple avec un texte et un rectangle. Nous faisons ensuite glisser une occurrence de ce bouton sur notre animation et nous l'appelons *photo*.

Nous en profitons pour faire glisser une autre occurrence de notre clip vidéo que nous appelons maintenant *fondphoto* : ce clip sera le réceptacle dans l'animation Flash de la photo prise.

Lors d'un clic sur le bouton, le code suivant (à placer avant la précédente instruction stop()) permet de récupérer l'image affichée par la webcam et de l'afficher dans l'animation (ce qui n'est pas obligatoire mais permet de contrôler le bon déroulement de cette étape) :

```
photo.addEventListener(MouseEvent.CLICK, prendrePhoto);

function prendrePhoto(event:MouseEvent):void {
    var image:Bitmap;
    var imageDatas:BitmapData;

    imageDatas = new BitmapData(480,360, false, 0);
    image = new Bitmap(imageDatas);

    // On copie la vidéo dans l'image (dans ses data en fait)
    imageDatas.draw(lavelideo);
    image.width=240;
    image.height=180;
```

```
// On ajoute la photo au clip
fondphoto.addChild(image);
}
```

## Etape 2 : encodage de la photo

Nous devons maintenant encoder notre image pour en faire un jpeg. CSS ne le permet pas nativement, nous allons donc ajouter l'extension corelib : <http://code.google.com/p/as3corelib/>.

Une fois cette extension téléchargée et décompressée, on peut reconstituer l'arborescence des sources dans notre projet et y ajouter tous les fichiers ou plus simplement copier le fichier `as3corelib.swc` dans le sous-répertoire `fr/Configuration/Components/Data` de l'emplacement d'installation de Flash et redémarrer le logiciel. On ajoute ensuite l'instruction d'import suivante au tout début de la page de code :

```
import com.adobe.images.JPEGEncoder;
```

Puis on complète la fonction :

```
// On encode l'image grâce à l'extension Flash
var jpegEncoder:JPEGEncoder = new JPEGEncoder(85); // Qualité
maximale, ou presque...
var jpegBytes:ByteArray = jpegEncoder.encode(imageData);
```

## Etape 3 : remontée vers le serveur

### Encodage de l'image

Nous avons jusqu'ici obtenu une image en jpeg à partir de l'affichage de notre webcam. Cette image va être remontée vers le serveur via une requête POST vers un script PHP qui se chargera de l'enregistrer sur le serveur. ActionScript 3.0 permet facilement d'interagir avec un script serveur par méthode GET ou POST, mais par contre le jpeg doit d'abord être encodé au format texte pour pouvoir être transmis. Pour ce faire, nous allons utiliser la méthode `base64` de codage/décodage de binaire vers texte. Pour plus d'informations sur le principe de Base64, on peut se reporter ici : <http://fr.wikipedia.org/wiki/Base64>

Là encore, ActionScript ne fournit pas directement les méthodes de codage, mais une simple recherche de Base64.as sur Internet permet de trouver une classe pour le faire. Avec la classe que nous utilisons, la méthode `prendPhoto` est complétée avec la ligne suivante :

```
var base64EncodedImg:String = Base64.encode64(jpegBytes, true);
```

### Envoi d'une requête POST

L'envoi d'une requête vers le serveur fait intervenir 2 objets :

- un objet `URLRequest` qui encapsule l'appel du script serveur
- un objet `URLVariables` qui représente les données à envoyer

Par analogie avec un formulaire HTML, on peut dire que ces objets correspondent respectivement à la balise `form` et aux champs `input`. On dirige dans cet exemple la requête vers un script PHP, mais il pourrait tout aussi bien s'agir d'une servlet ou d'un script dotnet. Voici le code, toujours à ajouter dans la méthode `prendrePhoto` (pour simplifier dans le cadre de l'exemple, car en ActionScript comme dans d'autres langages il est préférable de structurer et décomposer son code) :

```
// Requête vers le script photo.php sur le serveur
```

```
var sendDonnees:URLRequest = new URLRequest("photo.php");
sendDonnees.method = URLRequestMethod.POST;
// Les données à remonter
var variables:URLVariables = new URLVariables();
variables.laPhoto = base64EncodedImg;
sendDonnees.data = variables;
var loader:URLLoader = new URLLoader();
try {
    loader.load(sendDonnees);
    trace("Appel PHP OK");
} catch (error:Error) {
    trace("Erreur pendant l'appel PHP");
}
```

## Etape 4 : prise en charge par un script PHP

Le code du script `photo.php` est assez simple car nous avons juste à récupérer le contenu de la variable `laPhoto` transmis par Flash et à le décoder en Base64, PHP fournissant une fonction `base64_decode` qui permet de le faire directement :

```
<?php
// Les données envoyées par l'animation
$laPhoto = $_POST["laPhoto"];
// On sauve la photo
$fichier = fopen("laphoto.jpg", "w");
fputs($fichier, base64_decode($laPhoto));
fclose($fichier);
?>
```

Après exécution, vous devez retrouver le fichier `laphoto.jpg` sur votre serveur ; ce fichier doit bien entendu contenir l'image de la webcam.

Attention : il est bien entendu essentiel de publier le projet Flash et de transférer les fichiers HTML, Flash et PHP vers un serveur afin d'utiliser le protocole HTTP. Sans cela, l'appel au script PHP ne fonctionnera pas.

Par ailleurs, Flash impose comme restriction de sécurité qu'une animation ne peut appeler des scripts que vers le serveur dont elle est issue. Il faut donc placer tous les fichiers sur un même serveur avec un même nom de domaine.

En cas de non-fonctionnement :

- mettez à jour le lecteur Flash de votre navigateur ;
- vérifiez que le script PHP peut bien écrire sur le disque du serveur ;
- lancez l'animation au sein de l'outil de développement Flash et vérifiez qu'il n'y a pas de message d'erreur.

## CONCLUSION

Nous avons vu comment utiliser Flash pour pallier certaines limitations de HTML pour la remontée d'informations vers le serveur. L'avantage de notre méthode, c'est qu'elle s'appuie sur les technologies que vous avez l'habitude d'utiliser sur votre serveur. Il est bien entendu possible de réaliser les échanges dans les deux sens, et d'alimenter par exemple une animation Flash avec du texte ou même des images générées dynamiquement côté serveur.

### ■ David Doussot

Diplômé de l'INA-PG, certifié SUN Java  
Responsable formation JAVA-EGILIA

## LANGAGE

## Conception orientée objet en Java et C++ : une approche comparative



Difficulté : \*\*\*

Editeur : Pearson

Auteur : Franck Barbier

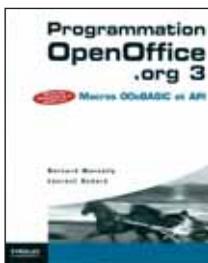
Prix : 27 €

La programmation objet, Java, C++ font beaucoup parler d'eux. Mais rien ne vaut une bonne

connaissance des deux langages pour savoir de quoi il retourne, évaluer les ressemblances et les différences. Le but est de poser les bases de l'objet dans les deux langages. On débute par une introduction générale à l'objet, puis rapidement, l'auteur aborde la programmation, les concepts, la pratique. Tout y passe : encapsulation, classe, composant, objet, héritage, polymorphisme, collection. Une bonne manière d'aborder l'objet.

## BUREAUTIQUE

## Programmation OpenOffice.org 3



Difficulté : \*\*\*

Editeur : Eyrolles

Auteur : collectif

Prix : 45 €

OpenOffice, c'est aussi une plateforme de développement et même du

lourd ! La preuve avec ce pavé de 1,6 kg et 960 pages ! Écrit par deux contributeurs majeurs de la communauté francophone d'OpenOffice.org, ce livre est devenu la bible du langage de macros OOoBASIC et de l'API de la suite bureautique. Il explique comment se servir de l'interface utilisateur liée aux macros et aux scripts afin d'automatiser des tâches répétitives, mais aussi comment tirer parti du langage de OOoBasic pour manipuler des documents, créer des boîtes de dialogue et des formulaires, exploiter des bases de données externes ou intégrées, intercepter des événements. Chaque point de l'API est décrit grâce à de nombreux exemples de macros et de routines réutilisables. Ce livre couvre aussi les nouveautés de la version 3.1 : Document-Properties, surlignement, notes et tableaux irréguliers dans Writer, gestion de diaporama dans

## LIVRE DU MOIS

## Sécurité PHP 5 et MySQL 2e édition

Difficulté : \*\*\*\* - Editeur : Eyrolles

Auteur : collectif - Prix : 32 €

Incontournable pour les développeurs PHP – MySQL, cet ouvrage revient en 2e édition. On ne change pas une formule qui marche. Les auteurs se penchent à nouveau sur l'ensemble des mécanismes de sécurité et ses problématiques. Le livre s'articule autour de 3 axes : les risques inhérents aux applications liées, les mesures de sécurité pour PHP et pour les technologies connexes. Écrit par deux experts ayant une pratique quotidienne de la sécurité sur la pile LAMP, ce livre recense toutes les vulnérabilités connues, les techniques pour s'en prémunir et leurs limitations. Très appliqué, il donne les clés pour se préparer à affronter un contexte complexe, où les performances, la valeur des données et la complexité des applications pimentent la vie des administrateurs responsables de la sécurité. A lire et à relire !



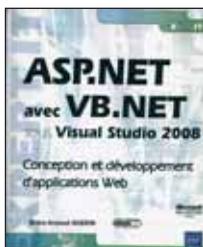
Impress, intégration des macros dans Base, jusqu'aux fonctions peu connues, sans oublier l'exportation PDF, l'export et import de texte encodé, le filtrage de données, en passant par les fonctions matricielles et la gestion d'événements...

## COLLECTION

## Collection Expert IT (éditions Eni)



JBoss (Franck Simon, 39 €) : JBoss est devenu en quelques années un serveur d'applications complet et pas toujours bien compris. Dans cet ouvrage l'auteur explique comment développer et déployer des applications JEE. L'ouvrage aborde l'architecture, les mécanismes de persistance, le déploiement, les EJB, la sécurité, etc.



ASP.Net avec VB.Net sous Visual Studio 2008 (B-A Guerin, 39 €) : l'auteur accompagne le lecteur dans une étude complète de la technologie ASP.Net et de Visual Studio 2008. Pour chaque thème abordé des exemples pratiques et utiles sont fournis en VB.NET. Le lecteur commencera par Visual Studio et ses outils (Refactoring, tests unitaires, Visual Source Safe,...) et par les évolutions du langage VB.NET. Le deuxième chapitre décrit le fonctionnement des applications IIS. L'ouvrage aborde aussi la supervision, Linq, les performances, Ajax.

## ECOLOGIE

## GreenIT

Difficulté : \*\*

Editeur : Eyrolles

Auteur : collectif

Prix : 25 €

L'informatique verte ou le greenIT constitue une priorité pour le monde informatique et tout particulièrement ce qui concerne la réduction de consommation énergétique. L'informatique représente plus de 2 % de la consommation mondiale !

Le Green IT, c'est-à-dire la réflexion qui va de la maîtrise de la dépense d'énergie au recyclage des déchets informatiques, est maintenant considéré comme un enjeu majeur pour les années à venir.

Cet ouvrage est un outil de prise de conscience et de réflexion sur les problématiques de développement durable appliquées aux systèmes d'information des entreprises. A découvrir d'urgence !

## SYSTÈME

## Windows 7 révélé



Difficulté : \*\*

Editeur :

Micro Application

Auteur : collectif

Prix : 27,50 €

En attendant la déferlante Windows 7, voici un ouvrage sympathique pour bien débiter en tant qu'utilisateur de Windows 7. Découvrez les petits secrets du système : interface, sécurité, IE 8, les performances... Rapide à lire il vous apportera tout ce qu'il est vital de connaître...

# Les outils des Décideurs Informatiques

*Vous avez besoin d'info  
sur des sujets d'administration,  
de sécurité, de progiciel,  
de projets ?  
Accédez directement  
à l'information ciblée.*

**L'INFORMATION  
SUR MESURE**



*Actu triée par secteur*

*Cas clients*

*Avis d'Experts*



*Actus*

*Evénements*

*Newsletters*

**L'INFORMATION EN CONTINU**



[www.solutions-logiciels.com](http://www.solutions-logiciels.com)



**BLU AGE** & **magicedraw™**  
AGILE MODEL TRANSFORMATION

présentent

A PARTIR DE  
**1490 €**  
PAR UTILISATEUR !

# M2 SPRING™ & M2 JAVA™

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

## BOOSTEZ VOS DÉVELOPPEMENTS D'APPLICATIONS SPRING & JAVA !

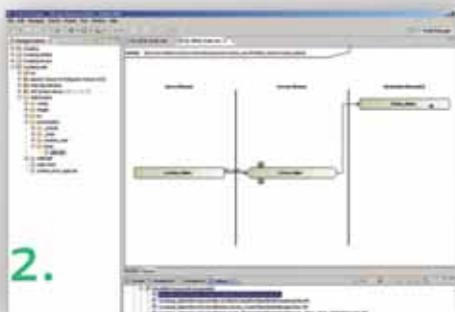
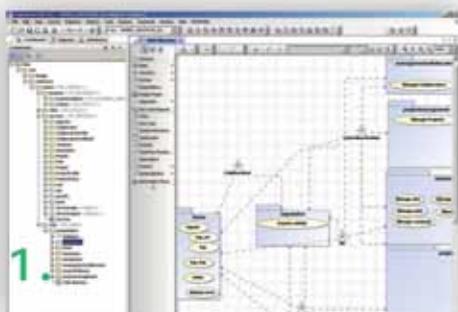
Egalement disponible :

### M2 FLEX™

**NOUVEAU!**  
- CRUD BOOSTER  
- CONSTRUIT SUR  
ECLIPSE 3.5  
(GALILEO)

Pour la génération automatique  
de vos applications riches !

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



Votre revendeur  
Comsoft-SOS Developers  
Tel : 08 25 07 06 07  
infos@comsoft.fr  
www.sosdevelopers.com

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

Informations et licences d'évaluation :  
**[www.model2code.com](http://www.model2code.com)** OU **01 56 05 60 91**

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