

Pro

Mensuel - Avril 2008 - N°107

LE MAGAZINE DU DÉVELOPPEMENT

grammmez!

www.programmez.com

WEB 2.0 RELOADED

► **Ruby on Rails 2.0** la "killer" technologie!

► **HTML 5.0** : la fin d'Ajax ?

Internet Explorer 8



Microsoft
Silverlight

Silverlight 2.0

vosre première application !



Développez vos applications mobiles avec **Android !**



En route vers Java 7

Découvrez le futur Java 7, les nouveaux frameworks - Toutes les nouveautés de Java 6 Update N - Quelle interopérabilité Java / .net ?

3D

Maîtriser les shaders

Vista

Maîtriser l'acquisition d'images avec l'API WIA

Java

Programmer le port série !

Linux

Visualiser les alertes de sécurité

DÉVELOPPEZ 10 FOIS PLUS VITE

INCLUS EN STANDARD DANS WINDEV 12 :

Environnement intégré (IDE & ALM)

- Tout est en **français**
- **Hot Line** personnalisée gratuite
- Déploiement libre et gratuit
- Création des **EXE** sécurisés, des Web Services, des applications **NET**, des applications **Java**...
- **32 bits, 64 bits**
- Compilation **JIT**
- Code multi-plateforme compatible **Mobile** et **Internet**
- Génération d'application **Java** à la demande
- Fonctionne en TSE et Citrix
- Générateur automatique d'**IHM**, avec charte graphique automatique. Création d'IHM «Vista» & «Vista Like» par utilisation de gabarits fournis
- Tous les Windows : 98, 2000, NT, XP, Vista, 2008...
- Générateur d'**Etats** et **Requêtes diffusables**, création de PDF, code-barres, étiquettes. Fond de page PDF
- Automatique dans chaque application: menu d'export vers Word, Excel, OpenOffice, XML, PDF, Graphiques 3D; Historique de saisie... Envoi de mail, Macros utilisateur
- **Real-RAD** : Générateur d'applications complètes
- **AAA** (Architecture Automatisée d'Application): créez votre propre RAD (Patterns)
- **Hyper File SQL**, Base de Données Client/Serveur, Locale et Mobile sous Windows et Linux (version illimitée, libre et gratuite); Gère 4 millions de Térés
- Accès à **toutes les Bases de Données** tierces: Oracle, AS/400, SQL Server, DB2, MySQL, Access, xBase, etc...
- **Reproduction** multibases assistée
- **XML natif**
- Accès natif à **SAP R/3**, Lotus Notes, LDAP, Outlook,...
- Centre de suivi du planning d'équipes
- Centre de suivi des retours utilisateurs
- **Tableau de Bord** de suivi de projet
- Centre de Modélisation **UML**, Merise et Souple; code généré depuis l'analyse, reverse engineering
- Dossier automatique : analyse & programmation
- Création et utilisation de **composants** ; 3-tiers
- Règles métier; Gestion des exigences
- Langage de 5^e Génération **LSG**, qui élimine 90% du code
- Ouverture aux **L4G** et **L3G**: C++, C#, Java, VB, Cobol...
- Import d'applications Access et VB
- Fonctions Domotique (norme X10)
- Gestion liaison série RS 232, parallèle et USB
- Fonctions Bluetooth,
- Fonctions réseau **SNMP**
- Fonctions TAPI, OPC, FTP, HTTP, Socket, Twain, API, DLL,...
- Editeur de code intelligent, avec test immédiat sans recompilation
- **Tests unitaires** de code et d'IHM, Editeur visuel de tests de non-régression
- Refactoring
- Outil de versionning sophistiqué
- Débugueur puissant: threads, composants, ... Débogage à distance
- **Profiler**, pour optimiser la vitesse du code
- Multilingue automatique: jusqu'à 20 langues
- Générateur d'aide **CHM**
- **Multimédia** intégré (image, son, vidéo)
- Générateur d'**Installations en 1 clic**, et mises à jour automatiques (local, à distance, via Internet)
- **Automatisation** en 1 semaine (manuel livré)

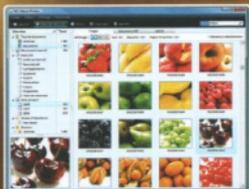
WINDEV® DÉVELOPPEMENT RICHE



RENDEZ-VOUS au TDF TECH 2008 PC SOFT

inscrivez-vous sur www.pcsoft.fr
du 27 mars au 29 avril, 11 villes

Développez de superbes IHM sans compétences graphiques grâce au générateur d'IHM (avec correcteur d'IHM) et aux gabarits fournis.



www.pcsoft.fr

Demandez le dossier gratuit (200 pages + 1 DVD)
Inclut la version Express (gratuite) et 100 Témoignages
détails. Tél: 04.67.032.032 ou info@pcsoft.fr

Fournisseur Officiel de la
Préparation Olympique



Logiciel professionnel. Documentation complète. Support technique gratuit. 15 jours pour le version en cours de développement.

> Actus

L'actualité en bref	6
Agenda	6

> Evénement

Le Service Pack 1 de Windows Vista	9
HTML5 : le langage qui va changer le visage du web	12
Mix OS : RIA et API web à l'honneur	18
Développer sa première application Silverlight 2	21

> SGDB

Développer une application C++ indépendante de la Base de Données	24
---	----

> Gros Plan : RUBY on Rails [1^{re} partie]

Ruby on Rails 2.0 : la nouvelle star du web !	26
Tout pour REST	31
Rails et la sécurité	35

> Carrière

École Centrale d'Électronique : miser sur l'embarqué !	38
--	----

> Cahier spécial Adobe

Adobe Flex 3 et AIR, une nouvelle ère pour le développeur	I
Adobe Flex 3 : une nouvelle gamme Open Source pour développer des RIA	II
Ma première application AIR avec Flex Builder	V
Adobe AIR : une nouvelle ère pour vos applications web	VI

> Dossier : En route vers Java 7

Les promesses de Java 7	49
Java 7 : en passant par Java 6 Update N	52
Les nouveaux frameworks du monde Java	55
Quelle interopérabilité Java / .NET ?	58
Java : un marché de l'emploi de plus en plus tendu ?	60

> Code

Android : mettez de l'Open Source dans votre téléphone	61
Les shaders et la 3D en temps réel	66
Visualisation de sécurité en Open Source	70
La programmation du port série du PC avec Java	73
Le langage Dada révélé	77
Maîtriser l'acquisition d'images sous Windows Vista	78

> Temps libre

Ludique	80
Les livres du mois	82



CD-Rom 107 PROGRAMMEZ !

JAVA

OpenJDK 6

Découvrez dès aujourd'hui la version open source de Java 6.

JEUX

Micropolis (Sim City)

Le code source complet du célèbre jeu Sim City. À vous de gérer votre ville !

Open Arena

Installez et découvrez Open Arena et les sources du moteur d'exécution 3D.

.NET / WINDOWS

Silverlight 2.0 bêta 1

La version 2 de Silverlight introduit de très nombreuses nouveautés (RIA, support .net et des langages dynamiques, etc.). Testez dès aujourd'hui la bêta 1 !

Internet Explorer 8 bêta 1

Découvrez dès maintenant le prochain IE ! Version bêta pour les développeurs et webmasters.

Unity

Unity est un conteneur léger d'injection de dépendance développé par l'équipe Patterns & Practices.

OPEN SOURCE

Gcc 4.3.0

La dernière version du compilateur C.

Drupal 6.1

CMS Open Source permettant de créer un site web sans une ligne de code.

PostgreSQL 8.3.0.1

La plus performante des bases de données open source.

Scala 2.7.0

Langage de programmation multi-paradigme conçu à l'EPFL pour exprimer les modèles de programmation courants dans une forme concise et élégante : Versions Windows et Linux.

OUTIL

RealBasic 2008 r1

Environnement de développement rapide (RAD) qui compile des applications natives pour Windows, Macintosh et Linux. Version complète de test valable 30 jours.

Arc

Arc est un nouveau dialecte Lisp conçu par Paul Graham.

Projet Domobox

Projet de fin d'étude de l'Epitech, pour automatiser la maison du futur. Versions Windows et Linux.

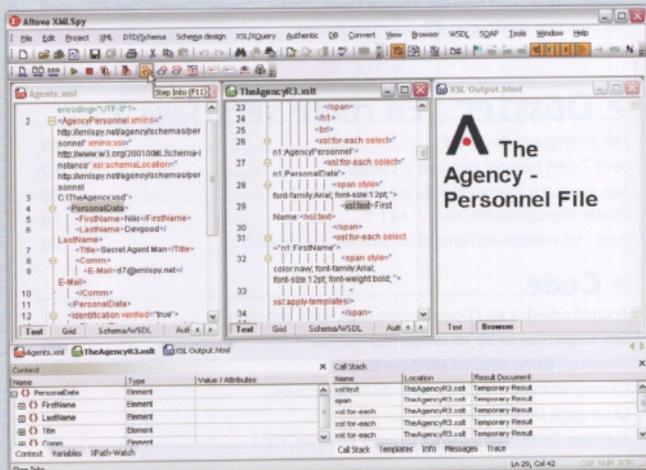


XMLSpy – Des outils pour les développeurs XSLT

Optimisez le développement de vos feuilles de style XSLT grâce à l'éditeur XML le plus populaire au monde. Altova[®] XMLSpy[®] vous offre une large palette de fonctions pour éditer, transformer, déboguer et profiler votre code XSLT :

- Prise en charge de XSLT 1.0 et Schema-aware 2.0
- Éditeur XSLT avec aides à la saisie intelligentes
- Fenêtre XPath Analyzer pour générer et tester des expressions XPath 1.0/2.0
- Neuf onglets dans XPath Analyzer pour stocker des résultats et faire des recherches sur plusieurs fichiers
- Développement XSLT pour OOXML dans les documents Microsoft[®] Office 2007
- Débogueur XSLT pour perfectionner vos feuilles de style
- Profileur XSLT pour optimiser vos feuilles de style
- Transformation instantanée et opérations par lots
- Vue Navigateur pour prévisualiser les transformations
- Et plus encore ...

Avec ses aides à la saisie XSLT avancées, ses assistants, ses débogueurs et ses profileurs, XMLSpy offre aux développeurs tous les outils qu'il leur faut pour créer les sites Web et les applications les plus sophistiqués du moment.



Découvrez pourquoi XMLSpy a été choisi par plus de 3 millions de développeurs XML.
Téléchargez une version d'essai gratuite de 30 jours sur www.altova.com



Le développeur ou le jardinier de demain

L'informatique verte ou green computing, en anglais, est d'actualité. Les constructeurs mettent à l'honneur le recyclage, la consommation énergétique. Depuis des années, on critique, avec justesse, la déperdition énergétique des ordinateurs. Nous sommes au début d'une évolution majeure de notre monde informatique : réduire la consommation électrique, mieux utiliser l'énergie consommée, dégager moins d'énergie.

On oublie peut être un maillon vital : le logiciel, donc le développeur. L'informatique verte ne peut pas se limiter au matériel. Le logiciel joue un rôle crucial : des technologies comme la 3D, les nouvelles interfaces, la compilation... exigent de la puissance qui nécessite elle-même de l'énergie. Comment faire mieux avec moins ?

On nous annonce 8, 12, voire 16 ou 20 cœurs sur un processeur d'ici 2010 / 2011. Les systèmes ont du mal à maigrir (peut être un bon coup de Wii Fit ?) et migrent sur les téléphones portables. La 3D s'impose, les nouvelles technologies du Web n'indiquent guère d'estampille * je suis vert, adoptez-moi ! *. Et a-t-on pensé à estimer la chaleur dégagée par notre développeur pour adapter son code à 12 cœurs ?

Plus sérieusement, si nous voulons aboutir à une informatique verte, il faudra passer le développement au vert : réduire drastiquement les ressources nécessaires, simplifier les technologies, les plates-formes, les langages, etc. L'informatique verte va-t-elle sonner le glas de l'application monolithique, des applications pesant plusieurs centaines de Mo ? Force est de constater que l'on n'en prend pas le chemin. Bien entendu il y a des API, des librairies pour mieux gérer la gestion de l'énergie, mieux accéder aux matériels. Mais, selon nous, il s'agit d'un palliatif. Il faut s'occuper du cœur du problème : la complexité logiciel- le toujours croissante. La simplicité implique-t-elle la complexité ? Ou la complexité appelle-t-elle la complexité ?

La guerre qui s'annonce sur tous les fronts informatique peut-elle aider l'informatique verte ? Cet affrontement mondial, a au moins du bon : plus de standards, un nouvel HTML capable de remplacer purement et simplement Ajax, plus d'ouvertures, plus d'interopérabilité. Et tous les éditeurs ont bien compris l'enjeu. Microsoft, contraint ou non, se lance à corps perdu dans cette ouverture car il y a un double jeu intéressant : réduire les critiques du monde libre, tout en permettant d'étendre l'influence de ses protocoles et donc de Windows.

Et si on parlait un peu du champ de bataille qui se prépare tranquillement sous nos yeux ? Incontestablement, le mobile - PDA et téléphone - est revenu au cœur des stratégies éditeurs. Google avec le projet Android cherche à pénétrer un marché colossal mais il est trop tôt pour en connaître l'éventuel succès, on pourra en tirer les premières conclusions tout début 2009. Microsoft est reparti à l'offensive en dévoilant officiellement le Silverlight Mobile. Mozilla veut y imposer son navigateur, Linux part en ordre dispersé. Palm tente de redorer son blason. Apple étend son influence vers les éditeurs et développeurs (100 000 développeurs pour accéder au SDK !), Nokia rachète Trolltech pour se renforcer dans le logiciel et l'interface. Et on attend la réaction d'Adobe depuis la sortie de AIR et de Flex 3.

■ François Tonic - ftonic@programmez.com



Programmez!

LE MAGAZINE DU DÉVELOPPEMENT

Rédaction : redaction@programmez.com

Directeur de la Rédaction : Jean Kaminsky

Rédacteur en Chef : François Tonic

Ont collaboré : F. Mazué, J. Vidames, J.M. Maman, T. Lebrun, J.B. Boisseau, L. Guillois, V. Perdureau, O. Thery, F. Levassaur, P. Chifflier, S. Tricaud, H. Darmet, C. Baret, F. Queudret, L. Juillard, R. Piacentini, N. Jacobus, B. Michel.

Dessin : Michel Piedoue, Jissey

Maquette : AJE Conseils

Publicité : Régie publicitaire, K-Now sarl

Pour la publicité uniquement : Tél. : 01 41 77 16 03
coordination@programmez.com

Éditeur : Go-02 sarl, 6 rue Bezout - 75014 Paris
Coordination@programmez.com - Dépôt légal : à parution - Commission paritaire : 0707K78366 - ISSN : 1627-0908 - Imprimeur : ETC - 76198 Yvetot

Directeur de la publication : Jean-Claude Vaudecrane
Ce numéro comporte 1 cd rom

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

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

PROCHAIN NUMERO

N°108 - Mai - Parution : 30 avril 2008

Le Devsigner arrive !

La fusion du développeur et du designer web est une réalité ! Quelles bonnes pratiques ? Quelles plates-formes et outils ?

Incontournable XML

Pourquoi XML est devenu la colonne vertébrale des données, des fichiers, des applications ?

Ruby On Rails (2^e partie)

IronRuby, intégration NetBeans et Rails, migration vers Rails 2

ART MODERNE

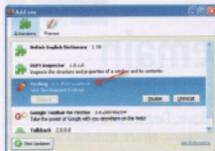
Souriez avec Jissey

www.programmez.com
chaque semaine,
un gif animé



- CULTURELLEMENT, JE TROUVE LES VERTICALES BEAUCOUP PLUS SUGGESTIVES QUE LES HORIZONTALES !

RUBY



Un nouvel IDE Ruby

Décidément les outils Rails sont à la mode. C'est désormais au tour d'Aptana de proposer sa solution Rails pour Aptana Studio, un IDE orienté développement web. L'outil est open source et librement téléchargeable. Pour l'éditeur, il s'agit de faciliter le développement Rails (éditeur, complétion de code, debug). Aptana a l'ambition d'être un IDE incontournable pour Rails, iPhone, AIR et PHP...

SERVEUR

Aptana sort un serveur Ajax

Aptana Jaxer est un serveur Ajax supportant HTML, Javascript, CSS, Ajax, JSON, DOM, etc. Le code peut s'exécuter sur le serveur, le client, ou les deux. Le développeur peut spécifier où s'exécute son code. Il s'insère dans un serveur Apache et d'autres serveurs web seront supportés comme IIS. Jaxer se base sur les technologies Mozilla. Le cœur de Jaxer est écrit en C / C++ mais on dispose d'un framework écrit en javascript qui permet ainsi d'étendre, de personnaliser le serveur. Avec Aptana Studio, l'éditeur propose une solution complète. Disponible pour MacOS X, Linux et Windows et open source. Le serveur Ajax sera-t-il la nouvelle tendance 2008 après le framework ?

INTEROPÉRABILITÉ

Microsoft lance l'initiative " document interoperability "

Décidément depuis quelques semaines, Microsoft surfe sur l'interopérabilité et l'ouverture. Après avoir dévoilé plusieurs dizaines de protocoles Windows, l'éditeur a lancé le 6 mars dernier une nouvelle initiative autour des documents. Il s'agit de promouvoir le choix du format de documents (dixit l'éditeur) pour le développeur, l'éditeur, l'utilisateur. Cette initiative a pour but de mieux informer, de faire une bonne interopérabilité entre les formats, de créer des templates, des tests d'interopérabilité, etc. Plusieurs éditeurs comme Novell, Mark Logic ont rejoint l'initiative. Un laboratoire à Cambridge est chargé de devenir le centre d'interopérabilité documentaire entre OpenXML et Open Document Format sur différents logiciels, systèmes, et terminaux. D'autre part, Microsoft a annoncé la version 1.1 de son convertisseur ODF - OpenXML pour Excel et PowerPoint.

INTEROPÉRABILITÉ

Microsoft : offensive sur l'interopérabilité et les communautés open source

Le 21 février marquera-t-il un tournant chez Microsoft et le monde informatique en général ?

Steve Ballmer, Ray Ozzie (architecte en chef), Bob Muglia (partie business sur les serveurs et outils), Brad Smith (partie juridique), ont fait part d'une annonce majeure, selon l'éditeur, autour de l'interopérabilité

pour les développeurs, les partenaires, la concurrence. Il s'agit d'une annonce importante même si elle suit le mouvement engagé par l'éditeur. L'initiative s'articule autour de 4 éléments : assurer la connexion " ouverture ", promouvoir la portabilité des données, améliorer le support des standards, enfin un engagement plus fort avec l'industrie et les clients, en incluant les communautés open source ! Cela passe par la disponibilité de 30 000 pages de documentation technique sur les API et protocoles. Il sera donc possible, à la place d'une décompilation ou d'autres méthodes, de savoir comment implémenter le protocole ou utiliser des API jusqu'à présent non documentées (ex. : réplication d'une base d'Active Directory). Comme le précise Bernard Ourghanlian de Microsoft France, le code source n'est pas inclus dans les annonces. Dans cette initiative, on retrouve : Vista, Office 2007, SharePoint Server 2007, Office 2007, SQL Server 2008, .Net, Windows Server 2008...

Bernard Ourghanlian précise aussi que l'annonce est plus axée entreprise que grand public. À noter que les outils de développement ne sont pas inclus dans l'annonce. Par contre, les versions futures des solutions citées plus haut seront documentées.

Pour une utilisation non commerciale par un développeur (ou d'une communauté open source), il n'y a aucune licence demandée par Microsoft sur un quelconque brevet. Par contre, pour une utilisation commerciale, il faudra acquiescer une licence. Pour aider l'interopérabilité entre Microsoft et les communautés ouvertes, l'éditeur lance Open Source Interoperability Initiative. Il y aura des ressources et des événements pour animer l'initiative. Les accords passés avec des éditeurs open source comme Talend, Zend ou encore JBoss ne sont pas caducs. On peut s'attendre à de nouveaux accords. À n'en pas douter, le monde informatique (et en particulier Open Source) réagira rapidement. Un argument de moins pour l'Europe et les réalisants d'une action légale contre Microsoft ? L'avenir nous le dira.

Agenda

AVRIL

- 1er, 2 et 3 Avril 2008, Paris Expo Porte de Versailles Hall 8 - **RTS EMBEDDED**, 16e salon des solutions informatiques temps réel et des systèmes embarqués - www.ni.com/embedded/fr/
 - Le 2 avril à Paris, palais Brongniard, **On Air Tour Europe 2008** Pour tout savoir sur AIR http://onair.adobe.com/schedule/cities/paris_fr.php
 - Le 8 Avril à Paris, réunion du **Groupe utilisateurs Java de Paris** " le box, les performances Java ". La session sera animée par Kirk Pepperdine (en anglais). www.parisjg.org
 - Le jeudi 10 avril à Paris, **Journées technologiques sur l'instrumentation**, organisées par National Instrument.
 - http://sins.ni.com/apps/utf8/miev.ni?action=display_offerings_by_event_id&event_id=24258&site=NIE&country=FR&node=162920&f
 - Les 15 et 16 Avril 2008, CNIT Paris La Défense. **COIP/VOIP 2008**. Toutes les communications sur IP ! <http://www.salon-coip.com/>
 - Les 15 et 16 Avril 2008 au CNIT Paris La Défense. **Séminaires Rencontres du Management de Projet**, pour faire le point sur les outils et méthodes www.groupe-solutions.fr/
 - Le 17 avril à Paris, Espace Pierre Cardin - **REMIX08**, une après-midi sur les technologies Web de Microsoft <http://www.microsoft.com/france/events/evenements.msp>
 - Du 27 mars au 29 avril 2008, **Tour de France PC Soft** : présentation de la V12 sur 11 villes <http://www.pcssoft.fr/pcssoft/tdftech/2008/index.html>
 - Du 8 au 29 avril, Tour de France dans 6 villes, **Objecteering SOA Solutions** www.objecteering.com
- ETRANGER**
- Du 22 Avril 2008 au 24 Avril 2008 Londres (Royaume-Uni) - Infosecurity Europe 2008 - **Salon international de la sécurité informatique**. www.redepxpo.com

JAVA

Eclipse 3.4 : pour juin !

La nouvelle version majeure, nom de code Ganyemed, est prévue pour fin juin prochain. Cette édition mettra à jour les différents projets Eclipse inclus. Le projet est développé sur un mix de JVM 1.4 et Java 5 ainsi que des fonctions spécifiques à Java 6 (donc JVM Java 6 requise). La compatibilité est assurée avec les projets venant d'Eclipse 3.x. Cependant, comme toujours, il faudra se référer aux guides de migration pour vérifier les points noirs. Sur la version MacOS X, des améliorations sur l'interface, le glisser-déposer. SWT fait son apparition sur Windows 64-bit Itanium, des annotations améliorées, le support du transfert URL (Drag and drop API), support OpenGL sous Solaris. D'autre part, les discussions autour du futur Eclipse 4.0 sont entamées depuis quelques semaines. EclipseCon 2008 aura été l'occasion de glaner idées et informations. Des réactions négatives sur la présence d'IBM (trop forte) sur le projet e4 (e4 pour Eclipse 4.0) secouent depuis mi-mars plusieurs communautés. Mais il est trop tôt pour dessiner les contours exacts de cette version qui a pour ambition de renouveler Eclipse en profondeur.

JAVA

NetBeans 6.1 : presque disponible

Encore en bêta, la v6.1 de NetBeans devrait sortir en version finale fin avril. L'IDE mise plus que jamais sur le développement avec un support plus intégré de javascript (complétion, analyses syntaxiques, refactoring). Spring est désormais en version 2.5. On bénéficie d'un support de

MySQL dans l'exploration de données. Ruby renforce encore sa présence avec des améliorations de son support. De nouveaux modules seront aussi disponibles : ClearCase, ASIX, SOAP UI. Sur la v7 pour le moment aucune information précise. Mais le support de Java 6 Update N pourrait être effectif avec sans doute des changements sur la partie Swing dans l'éditeur.

PHP

Des formations en français chez Zend

L'éditeur Zend propose depuis quelques semaines des formations entièrement en français ! Une bonne nouvelle ! Le programme E-learning de Zend Technologies couvre l'ensemble des axes de formation PHP – allant des principes de développement de base jusqu'aux cours pour développeurs expérimentés. De même, Zend propose des formations conçues pour répondre à certains aspects critiques ou à des publics spécifiques – " Sécurité en PHP ", " PHP pour programmeurs RPG " etc. Zend propose également des offres de formations " groupées " permettant de suivre un cursus complet de 3 formations complémentaires à un tarif préférentiel. Ce programme de formations en ligne, conçu et délivré par les experts Zend, vient s'ajouter à l'offre de formations PHP déjà proposées par Zend et ses partenaires en formule " salle de classe " en inter- et intra-entreprise. Site : <http://www.zend.com/fr/store/php-training/>



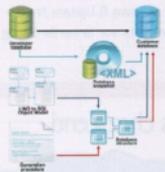
Microsoft Press préférez l'original !



Guide pratique de
Windows Server 2008

DONNÉES

Perpetuum Software aide vos données



L'éditeur russe a lancé Database Restyle – Application. L'outil permet de comparer et de synchroniser les structures de bases de données, facilitant le travail de maintenance ou de récupération quand on change de base. Il sert d'interface à l'outil déjà disponible : Database Restyle – Library (simple composant). Cette interface doit permettre d'éviter le traitement

manuel ou le moindre script. La structure peut être sauvegardée au format XML. Disponible pour SQL Server 2005.

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

PHP

Zend officiellement certifié Windows Server 2008

Zend Core de l'éditeur Zend est depuis début mars officiellement certifié sur Windows Server 2008, un gage de qualité et de disponibilité en environnement critique. " Microsoft et Zend continuent de collaborer pour assurer la performance et la fiabilité de PHP sous Windows " déclare Sam Ramji, Directeur de la technologie Open Source chez Microsoft. " Ensemble, nous créons une compatibilité de haut niveau et nous supportons la façon de travailler et les styles de développement que nos clients souhaitent utiliser sur Windows Server ". Depuis des mois, les annonces de collaboration entre Microsoft et Zend se sont multipliées : disponibilité de FastCGI (pour utiliser PHP dans IIS), support de CardSpace, bon fonctionnement d'une douzaine de projets phares du monde PHP comme Mabo ou WordPress.

RACHAT

W4 rachète Lyria

Surprise dans le monde Java et de l'interface homme-machine, l'éditeur Lyria, concepteur de Leonardi, vient d'être racheté par W4, éditeur spécialisé dans le BPM. Cette acquisition doit permettre à W4 de renforcer son offre logicielle, notamment sur la partie développement rapide d'applications métiers, mêlant ainsi la partie processus et la couche interface. " Je me réjouis de ce rapprochement qui va nous ouvrir de nouvelles perspectives de développement. En effet, en plus d'aider nos clients à automatiser leurs processus, nous allons leur permettre de les enrichir avec des interfaces prenant en compte l'ensemble de leurs besoins applicatifs " précise Laurent Hénault, Président de W4.

CYCLE

Le cycle de vie adopte le mode SaaS

Le logiciel + service s'impose aussi dans les solutions de gestion de cycle de vie. L'éditeur Serena va ainsi proposer courant 2008 : Mariner, Business Mashups et Agile Application Lifecycle Management. Le mode SaaS apporte de nombreux avantages aux entreprises puisqu'il n'y a ni infrastructure à gérer ni logiciel à installer. Mariner (gestion de porte-

feuille de projets) doit arriver en ligne courant avril, pour un prix de base de 18 dollars par utilisateur et par mois. Jeremy Burton, le PDG de Serena déclare : " L'Internet fournit tout simplement un meilleur moyen de mettre des logiciels à la disposition de nos clients. Ils n'ont pas à gérer d'infrastructure, ils sont rapidement opérationnels et ils ne paient que ce qu'ils ont utilisé en acquittant un abonnement mensuel. C'est l'avenir de l'industrie du logiciel ! Nous avons donc commencé à migrer tous nos systèmes Internets vers le modèle SaaS : Salesforce.com, MarketBright, Postini, et Facebook ne sont qu'un début pour nous ". Site : www.serena.com

OBJET

Numicom et Epitech travaillent ensemble

Les deux entités travaillent de concert sur le projet Majook. Majook est issu du GameDev Lab de l'Epitech. Il fournit aux étudiants les notions indispensables pour comprendre le développement de jeux et acquérir les méthodes de développement. Numicom propose aux étudiants du laboratoire de collaborer aux évolutions des drivers et kits de développement dans le cadre de Majook, contre une rémunération (dixit le communiqué). La plate-forme interactive, Majook, se présente sous la forme d'un plateau électronique permettant la détection de la position et la nature d'objets communicants, tels que des pions, des figurines de jeux à historique embarqué, des cartes à jouer intelligentes... Ce plateau communique avec un ordinateur et, bientôt, directement avec une télévision. Pour jouer, il suffit de charger ou télécharger une application et de disposer sur le plateau, le tapis de jeu et les figurines correspondant au game play. Pour plus d'informations : <http://gamedevlab.epitech.eu>, et <http://www.numicom.com>.

SÉCURITÉ

OpenLiberty veut plus de sécurité

OpenLiberty.org diffuse OpenLiberty-J. Il s'agit d'une librairie visant les applications Web 2.0 en s'appuyant sur des standards et offrant un niveau de sécurité élevé. OpenLiberty-J permet aux développeurs d'applications d'incorporer rapidement et facilement les fonctions de sécurité et de confidentialité de haut niveau. Fonctions issues du standard Identity Web Services Framework de Liberty Alliance à l'interopérabilité prouvée. Elles visent les applications consommatrices d'identité telles que celles présentes dans les architectures orientées services (SOA) d'entreprise, les environnements de réseau social Web 2.0 et les applications client sur les PC et appareils mobiles. Diffusé aujourd'hui en version bêta sous la licence Apache 2.0, le code d'OpenLiberty-J peut être examiné et téléchargé sur OpenLiberty.org. La librairie est disponible en Java. Site : <http://OpenLiberty.org>

PROCESSEUR

AMD et TopCoder misent sur les outils de multithreading

AMD, fondeur et TopCoder, spécialisés dans les compétitions de développement, annoncent plusieurs concours de développement multicoeur. Le concours a débuté le 12 mars dernier. En tout, 4 compétitions sont prévues ! A vous de jouer ! Site : <http://www.topcoder.com/tc?module=Static&d1=sponsors&d2=amdOverview>

Le Service Pack 1 de Windows Vista

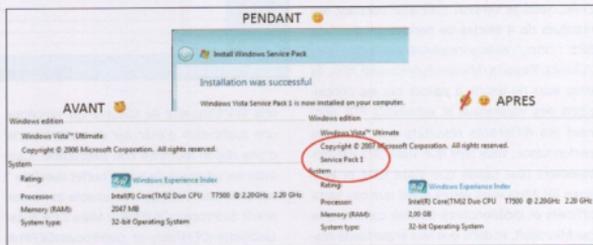
Les annonces se sont suivies et multipliées : Windows Server 2008 et le SP1 de Vista sont RTM – Release To Manufacture – au 4 février 2008. Disponibilité de Windows Server 2008 le 10 février 2008. Disponibilité du SP1 de Vista le 14 février 2008. Disponibilité de Vista SP1 le 25 février 2008. Disponibilité du SP3 RC2 de Windows XP le 27 février 2008... La numérogie n'étant pas d'une grande aide dans cette suite de nombres, nous avons décidé de creuser ces annonces pour comprendre ce que nous pouvons attendre du SP1 de Vista.

Depuis plusieurs mois maintenant, les heureux possesseurs d'un PC équipé de Windows Vista attendaient le Service Pack 1. Arrivé 1 an après la sortie mondiale de Vista, ce SP1 est censé apporter plus de performance, de fiabilité, de sécurité et quelques nouveautés. Le sujet étant d'importance, nous allons nous pencher sur l'installation du SP1 et vérifier que les promesses sont bel et bien tenues, mais aussi étudier ce que nous, développeurs, pouvons trouver comme nouvelles fonctionnalités ou évolutions qui pourraient impacter nos développements.

Tout commence par une "install" ? ... Non, un backup !

Pour installer le SP1 sur Windows Vista, vous avez le choix entre installer le SP1 (450Mo) ou faire un upgrade vers un Vista intégrant nativement le SP1 (compter environ 3Go). Le tout en 32 ou 64 bits, mais actuellement disponible uniquement pour la version anglaise, française, allemande, japonaise et espagnole. Les autres langues seront disponibles courant mars. Pour ma part, ayant depuis longtemps goûté aux saveurs de Vista, point d'upgrade mais plutôt une installation du Service Pack 1 en mode "stand-alone".

Des articles, news, blogs... sur l'installation de Vista – bêta, RC, RTM... – j'en avais vu passer comme tout le monde, et, à vrai dire, difficile de se faire une idée correcte sur l'installation du SP1. Pour certains, l'installation se déroulait sans problème alors que pour d'autres, leurs machines pouvaient tout simplement ne pas redémarrer, entre autres problèmes. Alors légende urbaine ou réalité ? Pas facile de discerner le vrai du faux tant que l'on en n'a pas fait soi-même l'expérience. C'est donc avec un peu de moiteur dans les doigts que j'ai lancé l'installation du SP1 sur mon Ultimate (qui jusque là fonctionnait sans problème). La première installation... comment dire...ne fut pas à la hauteur de mes espé-



rances (version " politiquement correct ") voire désastreuse (version réaliste) : aucun problème pendant l'installation du SP1 sauf qu'au reboot de ma machine, la barre de progression du chargement de Windows s'animaient mais c'était bien la seule chose animée qui tournait sur mon PC. Après diagnostic, le chargement de Windows bloquait en fait sur le fichier crosdisk.sys, un nom à vous donner des frissons dans le dos. Heureusement que j'avais pensé à faire des sauvegardes ; Malheureusement je n'avais pas de backup complet de ma machine (OS, applications...) et, encore plus affreux, j'étais l'heureux utilisateur d'un Vista OEM pré-installé sur mon portable Asus : pas d'alternative possible, me voila parti pour une réinstallation complète de ma machine avec le CD de récupération système livré par Asus (formatage de la partition primaire au passage).

1 jour plus tard et mes sauvegardes restaurées, j'étais revenu au point de départ. J'aurais pu abandonner, mais comme je suis persévérant (et surtout j'avais promis cet article), je me relance en prenant plus de précautions : backup complet et débranchement de tous les périphériques qui pourraient engendrer une instabilité. Quelques incantations plus tard et un suspens insoutenable digne d'un Hitchcock (il faut compter 1 heure d'installation), mon Vista Ultimate redémarrait avec le SP1. Moralité : Avant d'installer le Service Pack 1 de Windows

Vista, ton système entier tu backuperas. Tout le monde le sait, tout le monde le dit, mais est-ce que tout le monde le fait ?

Promesses tenues ?

Nombre d'utilisateurs de Windows Vista s'étaient plaints de la lenteur globale en se référant à Windows XP SP2. Il était évident, et Microsoft ne s'en est jamais caché, que la copie de fichiers, l'exécution de certaines applications, etc. étaient plus lents. Comme je m'étais informé pour l'installation du Service Pack 1 de Windows Vista, j'ai continué mon enquête en croisant plusieurs sources d'informations pour tenter de trouver la vérité sur les améliorations apportées par le SP1. Au final, pas de réponses tranchées ; La vérité doit encore être ailleurs mon cher Mulder... Pour aller au-delà des rumeurs et trouver les réponses par moi-même j'ai donc cherché des benchmarks sur le sujet :

- XP SP2 vs. Vista RTM vs. Vista SP1: Gaming benchmark – Source : <http://blogs.zdnet.com/hardware/?p=1390&page=3>. Conclusion : XP SP2 plus rapide, Vista RTM vs Vista SP1 pas de différences.
- Vista 32-bit vs. Vista 64-bit – Benchmark – Source: <http://blogs.zdnet.com/hardware/?p=1354&page=2>. Conclusion: Vista SP1 serait moins rapide que la version RTM?!
- Comparaison des performances entre Vista

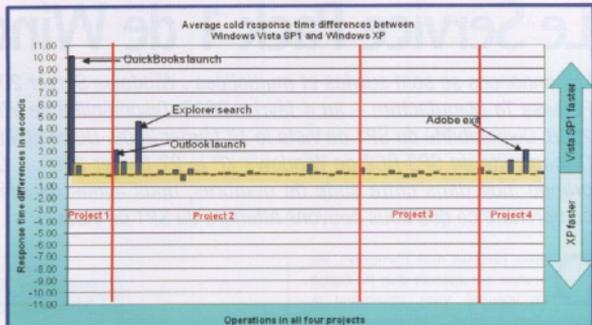
RTM et SP1 – Source: <http://4sysops.com/archives/vista-sp1-performance-comparison-with-vista-rtm/>. Résultat : Pas de différence pour 4 benchmarks sur 5. Le dernier faisant apparaître 10% de gain de performance pour Vista SP1 dans les scénarii bureautiques.

A croire que Vista SP1 est moins rapide que Vista RTM! 1 an d'attente, 450 Mo de téléchargement et 1 heure d'installation plus tard, j'aurai donc un OS Vista SP1 moins performant que sa version RTM qui, elle-même, était déjà moins performante que son prédécesseur XP ? Enfin, voici la version officielle donnant les résultats de 4 études de benchmark de Vista SP1 : <http://www.principledtechnologies.com/Clients/Reports/Microsoft/microsoft.htm>. Si vous avez du temps à passer sur les conclusions des volumineux et exhaustifs PDF donnant les différents résultats des tests de performance, vous lirez que Vista SP1 est globalement plus rapide que Vista RTM et Windows XP. Mais vous verrez aussi que ces tests officiels et indépendants ont été commandés par Microsoft, et donc que leur impartialité restera sujette à discussion par les détracteurs de Vista.

Quoi de neuf sous le capot ?

Vista RTM a apporté beaucoup d'innovations: IHM/Aero, .NET framework 3.0 natif, plus de 7000 API nouvelles, sécurité renforcée, etc. Alors que reste-t-il au SP1 pour encore nous étonner ? Des progrès et améliorations ont été apportés au niveau de l'infrastructure (déploiement, sécurité, performances, support de nouveaux hardwares et de nouveaux standards...) mais il existe aussi quelques nouveautés pour les développeurs :

- Support de la version 64 bits de MSDDL (la passerelle OLEDB pour les drivers ODBC), ce qui simplifie grandement la migration d'applications de plates-formes 32 bits à Windows Vista 64 bits.
- Support de Direct3D 10.1, mise à jour de la version 10 et qui étend l'API pour supporter les fonctionnalités des cartes graphiques de dernière génération. Ce qu'il faut retenir : plus de performances et plus de qualité. Voir les références pour accéder aux multiples PDF écrits sur le sujet à l'occasion du SIGGRAPH 2007 (Special Interest Group for Computer GRAPHICS).
- De nouvelles API Win32 qui permettent de mieux contrôler les paramètres liés au Data Execution Prevention (DEP) pour les applications. Le " Data Execution Prevention " est



une fonctionnalité de sécurité qui empêche une application d'exécuter du code à partir d'une région mémoire non exécutable. Cela évite les attaques de type " buffer overflow " par exemple. L'API incontournable et nouvelle est `SetProcessDEPPolicy`. Mais il y a aussi `GetSystemDEPPolicy` et `GetProcessDEPPolicy`. Très utiles pour des applications développées avec de vieilles versions d'ATL ou pour contourner des problèmes de compatibilité avec DEP sans avoir à désactiver ce dernier.

• Evolution d'Internet Information Server (IIS) 7 sur Vista SP1 aligné maintenant sur IIS 7 de Windows Server 2008. Améliorations de la performance et de la stabilité. Le support de FastCGI est maintenant natif et cela permet d'intégrer PHP par exemple.

Pour avoir la liste exhaustive de toutes les améliorations apportées par le Service Pack 1 de Vista, je redirige le lecteur vers le document d'introduction accessible en ligne sur : <http://www.microsoft.com/downloads/details.aspx?familyid=39B802EA-B2CF-4585-8CEA-2CC6A6247CB3&displaylang=en>. Ne vous laissez pas décourager par la première page : la photo d'un homme devant son PC avec Windows Vista et l'impression qu'il est derrière les barreaux. Serait-il prisonnier... ?

Conclusion

Le Service Pack 1 de Windows Vista a apporté beaucoup de correctifs et d'améliorations par rapport à la version RTM notamment au niveau de la performance, du support de nouveaux périphériques et de nouveaux standards. Le SP1 est un condensé des correctifs et patches développés par Microsoft depuis la sortie de l'OS et, comme tout service pack (excepté le SP2 de Windows XP), n'a pas vocation à appar-

ter de nouvelles fonctionnalités. Malgré tout, la disponibilité de Windows Server 2008 à la même date que le SP1 de Vista a permis de bénéficier des améliorations implémentées dans la version serveur. La question finale est : ces améliorations profiteront-elles aussi au SP3 de Windows XP ? Ce qui risquerait encore d'alimenter le débat entre " rester sur XP SP3 " ou " passer à Vista SP1 " ou encore le débat " je passe à Windows Server 2008 sans passer par Vista ". Quoi qu'il en soit, tout cela est affaire de besoins personnels et de puissance sous le capot de votre machine. Pour moi, le choix est déjà fait : Vista Ultimate SP1 !

Références

- Windows Vista blog : <http://windowsvista-blog.com>
- Présentation de Direct3D 10 et 10.1 pour le SIGGRAPH 2007 (35 Mo) : <http://www.microsoft.com/downloads/details.aspx?FamilyID=96cd2845-4c15-475e-a2dc-1d37f67fa6cd&DisplayLang=en>
- DEP sur le Web Log de Michael Howard : http://blogs.msdn.com/michael_howard/archives/2008/01/29/new-ix-apis-added-to-windows-vista-sp1-windows-xp-sp3-and-windows-server-2008.aspx



■ Frédéric QUEUDRET

CTO de la société Mpower, accélérateur d'innovation. Société française d'édition de logiciels et de prestations de services sur la création d'outils de productivité pour les développeurs Visual Studio. <http://www.mpower.com/>



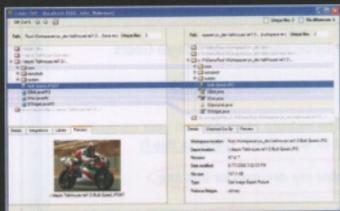
La fonctionnalité Folder Diff, un atout de productivité du système de GCL Perforce.

Folder Diff est un outil interactif d'affichage en juxtaposition permettant de comparer l'état de deux groupes de fichiers.

À l'aide de Folder Diff, on peut rapidement déterminer les différences entre les fichiers situés dans des dossiers, des branches, des étiquettes ou sur votre disque local. Cette fonction est particulièrement utile lorsque vous devez réaliser des fusions de codes complexes.

De plus, si vous travaillez deconnecter, Folder Diff facilite la synchronisation des données avec celles du serveur Perforce lorsque vous vous connectez de nouveau au réseau.

Folder Diff n'est qu'un des nombreux atouts de productivité offerts par le système de GCL Perforce.



Folder Diff de Perforce

HTML5 : le langage qui va changer le visage du web

Depuis la création du web au début des années 90, HTML n'a cessé d'évoluer tant au niveau de ses spécifications que de ses implémentations dans les navigateurs. Tout au long de son histoire, HTML est en effet sans cesse tirailé entre plusieurs univers aux approches contradictoires : celui des scientifiques (HTML a été créé au CERN), celui des bidouilleurs (les étudiants de quelques universités l'ayant popularisé), et celui des éditeurs de logiciels (serveurs, navigateurs, éditeurs HTML...).

Les éditeurs de logiciels ont donc proposé régulièrement des implémentations d'HTML qui devaient séduire les bidouilleurs à tout prix, y compris au risque de s'éloigner de standards et d'une logique scientifique. Avec le temps, chacun s'est rendu compte des problèmes de cette tendance sur le long terme et a promis plus ou moins de se rapprocher des standards. HTML 4, XHTML 1.0 et DOM HTML 2 (qui définit une API d'accès à HTML pour Javascript) ont donc été des efforts appréciables dans cette bataille pour plus de cohérence...malheureusement, ils s'avèrent largement insuffisants, en particulier en termes de fonctionnalités, pour les besoins d'un web moderne. HTML4 datant de 1997 et XHTML2 prenant une tournure qu'elles n'appréciaient guère, quelques personnalités du web (venant en particulier du monde des navigateurs), formèrent le WHAT Working Group pour travailler sur une spécification qu'ils nommèrent simplement HTML5. En mai 2007, le W3C finit par admettre le bien fondé des travaux du WHAT WG et accepta d'intégrer HTML5 à ses travaux de normalisation.

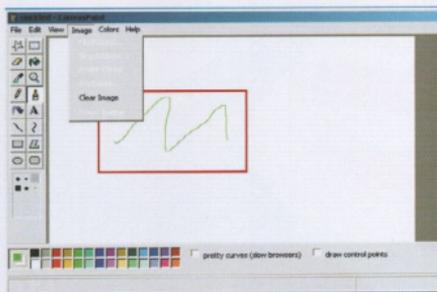
HTML5 dans les grandes lignes

Le WHAT WG s'est voulu pragmatique dans son approche, sacrifiant souvent la beauté du design aux demandes des utilisateurs... ce qui a eu pour effet de déclencher des débats houleux où les partisans d'XHTML 2.0 montraient du doigt la trop grande permissivité d'HTML5. Peu importe le bien fondé de ces débats puisqu'ils sont désormais d'arrière-garde : HTML5 verra le jour et nombreux seront les navigateurs à en implémenter les spécificités. Le pragmatisme d'HTML5 se retrouve dans quelques principes de conception marquants :

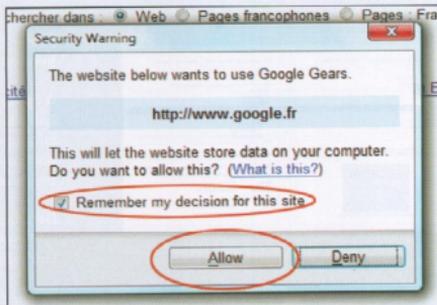
- une compatibilité maintenue avec le contenu HTML existant
- une norme définissant à la fois HTML5, XHTML5 (qui n'est que sa transposition dans la norme XML) et DOM HTML afin que toutes les balises et leurs comportements dynamiques soit pensés en cohérence
- une priorité clairement donnée à la satisfaction des utilisateurs ainsi qu'aux auteurs de contenus
- la volonté de répondre à toutes les problématiques du web : du contenu statique aux applications multimedia dynamiques les plus complexes.
- la multiplication des balises afin d'avoir une sémantique plus précise et plus efficace sur le long terme

Et concrètement ?

Concrètement, l'impact d'HTML5 va surtout se manifester dans les nouvelles fonctionnalités que les navigateurs web vont proposer à leurs utilisateurs de manière native, c'est à dire sans plug-in ni code javascript



Paint.net, un clone web de paint utilisant massivement canvas



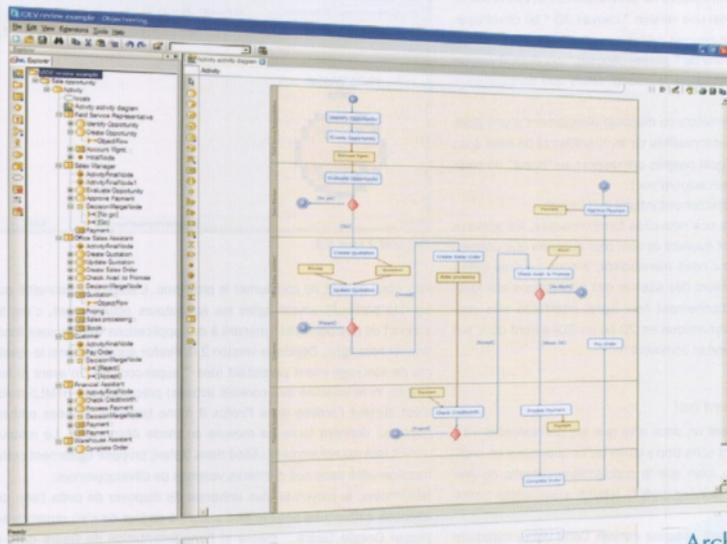
Google Gears permet le stockage local... mais vous demande votre avis !

compliqué. Citons les fonctionnalités les plus marquantes :

- les balises audio et vidéo, que les navigateurs devraient traiter sans faire appel à la foule des plug-in existant à ce jour (Windows Media Player, Quicktime, Realplayer, Flash...) et qui, surtout, devraient être pilotables en javascript via une API appropriée.
- une API pour le offline : tout comme aujourd'hui avec le plug-in Google Gears, il sera possible de prévoir une utilisation hors ligne des applica-

Your projects deserve a tool*

Objecteering Modeler



Objecteering 6.1

guide vos développements par les modèles sur toute la portée du système

Objecteering 6.1 est une solution de modélisation complète et ouverte supportant les standards OMG de modélisation UML 2.1, BPMN, et SysML, la modélisation de l'Architecture d'Entreprise et la modélisation d'une architecture SOA.

Il intègre le support de l'analyse des besoins et de la définition du dictionnaire, assurant ainsi une traçabilité complète sur tout le cycle de vie.

La technologie MDA associée à une ouverture de l'outillage en Java permet de guider le

développement à chaque contexte utilisateur, chaque infrastructure technique cible. Ses générateurs sur étagère automatisent le développement d'applications Java, J2EE, .Net C#, C++, SQL...

Objecteering Software, éditeur de l'atelier Objecteering 6, est le spécialiste français UML/MDA pour le développement d'applications guidé par le modèle. Son offre modulaire couvre le cycle de vie de la gestion des exigences jusqu'au déploiement d'application.

Pour plus d'information sur Objecteering 6.1, pour télécharger Objecteering **Free Edition** ou Objecteering **Enterprise Edition**, rendez-vous sur : www.objecteering.com

Architecture
d'Entreprise

UML2 BPMN
C# MDA SysML
Java SQL
J2EE SOA
C++

Objecteering
SOFTWARE

Venez découvrir
Objecteering
SOA Solution

Tour de France en avril
Séminaires gratuits

Inscrivez-vous sur
www.objecteering.com

ou contactez-nous au
01 30 12 16 60

* Vos projets méritent un outil

Événements

tions web incluant notamment un stockage de données sur le poste client via des "super-cookies" ou une base de données embarquée.

- une balise canvas permettant de dessiner dynamiquement à l'aide de javascript. Si la spécification actuelle ne couvre que la 2D, elle laisse la porte ouverte à la 3D : ainsi une version "canvas 3D" en développement chez Mozilla et Opéra pourrait même exploiter OpenGL.
- une API pour gérer les actions de "glisser-déposer" qui étaient jusque là réalisées via des bibliothèques Javascript telles que Scriptaculous ou JQuery.
- une balise datagrid qui permettra de disposer directement d'une grille de données munies de fonctionnalités de tri, d'édition et de liens à un fournisseur de données : quel progrès par rapport au "bazar" de code que représente un tel widget aujourd'hui !
- un système de template directement intégré au langage

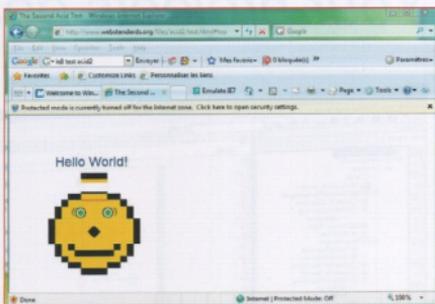
Comme on le voit à travers ces nouvelles fonctionnalités, les apports d'HTML5 sont véritablement majeurs et sont plus concrets que conceptuels. Tous les éléments qui nous manquaient, à nous pauvres petits développeurs web, pour réaliser des sites et des applications aux fonctionnalités avancées (fonctionnement hors ligne, interfaces très enrichies, multimedia, dessin dynamique en 2D ou en 3D) seront donc bel et bien là... reste à voir quand et comment !

Les fonctionnalités disponibles dès aujourd'hui

HTML5 n'est pas seulement un doux rêve que les développeurs web peuvent caresser, imaginant sans trop y croire qu'ils pourraient en profiter dans 5 ou 6 ans. Non, bien que la spécification actuelle ne soit qu'un brouillon de travail ("working draft"), HTML5 a une réalité exploitable dès aujourd'hui.

L'exemple le plus frappant est la balise canvas. Cette balise introduite à l'origine par Apple pour contourner ses difficultés à implémenter la norme de dessin vectoriel SVG a rapidement séduit : elle est aujourd'hui supportée dans les principaux navigateurs "non-Microsoft", c'est à dire Firefox >1.5, Safari >2.0, Opera >9.0 et Konqueror >4.

L'intérêt majeur de cette balise, c'est de pouvoir dessiner dynamiquement sur le web sans devoir faire appel à Flash : une véritable petite révolution disponible depuis déjà quelque temps et pourtant encore assez peu utilisée... Et si l'absence de support de cette balise par Internet Explorer vous rebute, sachez qu'une bibliothèque javascript, *iecan-*



Test acid 2 sous IEB.

vas, nous permet de contourner le problème. L'autre fonctionnalité qui semble particulièrement agiter les navigateurs actuellement, c'est le support de fonctionnalités menant à des applications web pouvant fonctionner hors ligne. Depuis la version 2.0, Firefox supporte ainsi le système de stockage client persistant (des "super-cookies" n'ayant ni les limites, ni la volatilité des cookies actuels) préconisé par HTML5 mais c'est surtout l'arrivée dans Firefox 3 d'une base de données embarquée qui donnera toute sa mesure au mode déconnecté. Le moteur webkit (qui est notamment utilisé dans Safari) propose également cette fonctionnalité dans ses dernières versions de développement.

Néanmoins, le moyen le plus universel de disposer de cette base de données embarquée semble être encore à ce jour de s'en remettre au plug-in Google Gears... même si l'implémentation de Gears diffère quelque peu de celle préconisée par le W3C.

Les fonctionnalités disponibles demain

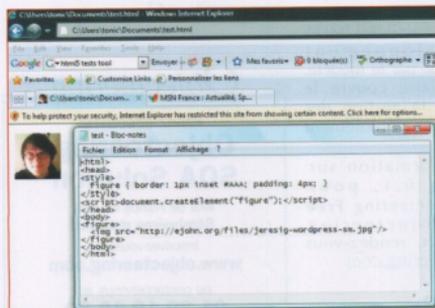
HTML5 étant une initiative des équipes de Mozilla, Safari et Opera, on peut s'attendre régulièrement à voir de nouvelles spécifications intégrées dans chacune de leurs versions de développement. Regardons où nous en sommes à ce jour.

Les balises audio et vidéo sont à l'état de développement avancé dans chacun de ces navigateurs. Le débat porte aujourd'hui sur les codecs que les navigateurs devraient embarquer nativement pour assurer la plus grande portabilité possible aux concepteurs de sites décidant d'utiliser ces balises.

Les formats Ogg Vorbis pour l'audio et Ogg Theora pour la vidéo paraissent les plus évidents dans la mesure où il sont les moins problématiques en termes de licences.

Moins sexy, mais tout aussi intéressant pour l'avenir du web, les fonctions de "cross-document messaging" sont également bien avancées : il s'agit de pouvoir échanger des informations entre sites web et dépasser ainsi une des limites de sécurité actuelles d'HTML. En effet, s'il paraissait logique d'éviter le "cross-site scripting", il était frustrant, à l'heure des flux RSS et des web services, de ne pouvoir permettre les interactions entre des pages web "non-hostiles".

Petit exemple : une page A, sur le domaine *monexemple.com*, contient un élément "object" qui lui-même embarque une page B, la page A pourrait alors envoyer un message à B avec un code Javascript du genre :



Exemple d'un code HTML5 s'exécutant sous IEB.

LEONARDI

LYRIA REJOINT LE GROUPE W4 !
L'UNION DU BPM ET DE L'IHM MD...
A SUIVRE...



L'ihm

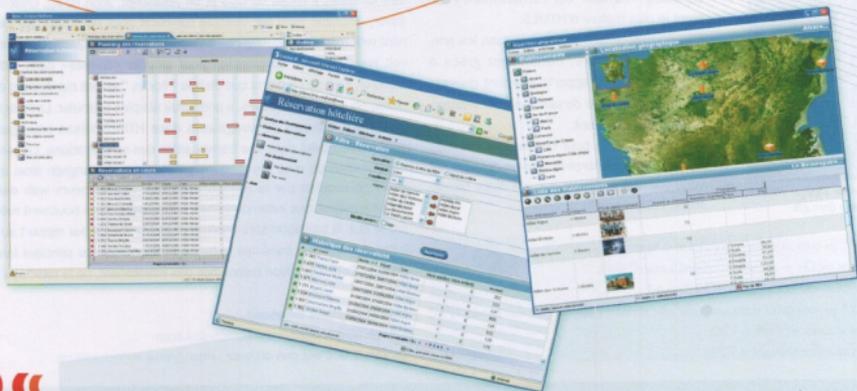
EN TOUTE SIMPLICITÉ !

Votre application de gestion multilingue avec plus de 100 vues de 20 types différents, en DHTML/Ajax, en Swing ou en plugin Eclipse, connectée à un SGBD et un bus JMS.

“ il vous faut
combien de temps
pour la réaliser ? ”

NOUVELLE
VERSION!
LEONARDI
V4.0
OPEN SOURCE

Si votre réponse est moins d'une semaine, inutile de vous rendre sur notre site, ni de télécharger la version gratuite de LEONARDI, sinon il est temps de passer à la vitesse "Model-Driven"...



LYRIA

Pour en savoir plus sur la solution LeonardI, rendez-vous sur notre site www.lyria.com ou envoyez-nous un courriel à info@lyria.com

Événements

```
var o = document.getElementsByTagName('object')[0];
o.contentWindow.postMessage('Bonjour page B');
```

La page B pourrait alors traiter l'événement de réception du message pour lui répondre ou effectuer un autre traitement :

```
document.addEventListener('message', receiver, false);
function receiver(e) {
  if (e.origin == 'http://monexemple.com') {
    if (e.data == 'Bonjour page B') {
      e.source.postMessage('Salut page A!', e.origin);
    } else {
      alert(e.data);
    }
  }
}
```

Et encore bien d'autres choses à l'avenir...

La place me manque malheureusement pour vous faire part de l'ensemble des nouveautés d'HTML5 (près de 450 pages...) et de leur état d'avancement dans les différents navigateurs. J'ai en particulier fait l'impatte sur les balises de formulaires (Web Forms 2.0) et sur des balises de structures documentaires (header, footer, nav, section...) Néanmoins quelques ressources web vous permettront d'aller plus loin :
- la version de travail du W3C : <http://www.w3.org/html/wg/html5/>
- une vue d'ensemble des différences entre HTML4 et HTML5 : <http://www.w3.org/TR/html5-diff/>
- la liste des spécifications implantées dans les principaux navigateurs : http://en.wikipedia.org/wiki/Comparison_of_layout_engines_%28HTML_5%29

Pourquoi et comment se mettre à HTML5 dès aujourd'hui ?

Je l'ai écrit à plusieurs reprises : HTML5 est d'ores et déjà une réalité tangible qu'il serait dommage de ne pas exploiter. Néanmoins, il est évident qu'il convient de prendre quelques précautions concernant une spécification qui est encore à l'état de brouillon. Voici donc quelques conseils quant aux fonctionnalités à explorer prioritairement.



Notre dessin réalisé avec dojo.gfx

Prêts pour canvas

La balise <canvas> est certainement l'élément le plus mature d'HTML5.

Elle est maintenant disponible dans les principaux navigateurs, notamment grâce à ExplorerCanvas et apporte vraiment quelque chose au web : du dessin vectoriel manipulable dynamiquement.

Je vous conseille pour commencer de vous appuyer sur une librairie comme dojo.gfx qui vous permettra d'avoir des résultats rapidement. Petit exemple :

```
/* je crée une div que j'ajoute à la page */
var maDiv = document.createElement("div");
document.body.appendChild(node);
/* je vais créer dans cette div une surface pour y créer des objets */
var surfaceWidth = 120;
var surfaceHeight = 220;
```

```
var maSurface = dojo.gfx.createSurface(maDiv, surfaceWidth, surfaceHeight);
var monGroupe = maSurface.createGroup();
/* je y vais dessiner un rectangle bleu... */
var monRectangle = { x: 100, y: 0, width: 100, height: 100 };
var monRectangleBleu = monGroupe.createRect(rectangle)
    .setFill([0, 0, 255, 0.5])
    .applyTransform(dojo.gfx.matrix.identity);
/* et maintenant un cercle vert ! */
var monCercle = { cx: 150, cy: 160, r: 50 };
var monCercleVert = monGroupe.createCircle(monCercle)
    .setFill([0, 255, 0, 1.0])
    .setStroke([color: "black", width: 4, cap: "butt", join: 4])
    .applyTransform(dojo.gfx.matrix.identity);
```

Le offline sans attendre

Les fonctionnalités de stockage hors ligne peuvent être aussi envisagées dès aujourd'hui, même si peu de navigateurs les implémentent. Je vous conseille en particulier de vous pencher sur la librairie dojo.storage qui vous permettra d'utiliser les possibilités, souvent méconnues, de stockage persistant de Flash : vos fonctionnalités "offline" seraient grâce à ce script potentiellement actives sur 98% des navigateurs... et ce dès aujourd'hui ! Si vous voulez aller plus loin et utiliser du SQL côté client, il nous faudra alors imposer à vos utilisateurs l'installation de technologies RIA telles que Google Gears ou Adobe Air : une condition qui est tout à fait envisageable dans le cas d'une application d'entreprise où le "parc client" est maîtrisé. Jack Slocum, leader de la librairie ext, nous donne ainsi un superbe exemple d'application web de gestion de tâches fonctionnant en mode offline indifféremment sous Air ou Gears (jackslocum.com/blog/2007/06/29/ext-air-and-red-bull). Il est néanmoins regrettable que les interfaces de ces technologies diffèrent légèrement de leurs équivalents HTML5...

La transition vers HTML5

A mesure que nos navigateurs préférés supporteront de nouvelles balises, il va devenir possible, du moins pour un certain nombre d'entre elles, de s'en servir sans attendre un support universel. En effet, comme les navigateurs peuvent interpréter du code ou des styles sur des balises qui leur sont inconnues, la plupart des nouvelles spécifications pourront être simulées à coups de CSS et de Javascript. John Resig, créateur de JQuery, explique ainsi par exemple comment nous pourrions dès maintenant redéfinir la balise <figure> à l'aide de styles adéquats. Or, quand on voit les difficultés surmontées pour simuler <canvas> dans Internet Explorer, il y a fort à parier que de tels "hacks" seront rapidement disponibles pour d'autres balises, a priori plus simples à émuler. La forte implication des éditeurs de navigateurs dans HTML5 laisse présager une certaine réactivité dans leur transposition des spécifications. La compatibilité maintenue avec les versions précédentes du langage ainsi que la maturité acquise par la communauté des développeurs web dans le contournement des incompatibilités entre navigateurs poussent même à croire que la transition sera finalement assez simple par rapport au saut technologique que constitue cette nouvelle version du principal langage du web. Cette transition commence dès maintenant, ne la ratez pas !

■ Jean-Baptiste Boisseau

Eutech SSII : <http://www.eutech-ssii.com>

Le web 2.0, c'est pas du buzz : <http://blog.eutech-ssii.com>



Pourquoi peindre avec les doigts ?

Visualisation Java pour clients riches et Ajax

ILOG JViews 8.1, la dernière version de la suite d'outils graphiques Java d'ILOG, couvre l'ensemble des fonctionnalités de visualisation avancée.

ILOG JViews 8.1 offre :

- Des composants graphiques puissants : diagrammes, courbes, tableaux de bord, cartographie, diagrammes de Gantt
- Des services évolués : agencement automatique de graphes, affichage performant pour des jeux de données volumineux

- Plusieurs techniques de déploiement : clients riches, applications Web interactives Ajax, Eclipse/RCP et portails
- Une expertise prouvée dans les industries les plus exigeantes : Informatique, télécoms, transport, énergie et défense.

Testez un de nos produits Java dès aujourd'hui <http://jviews.ilog.com>



Changing the rules of business™

Mix 08 : RIA et API web à l'honneur

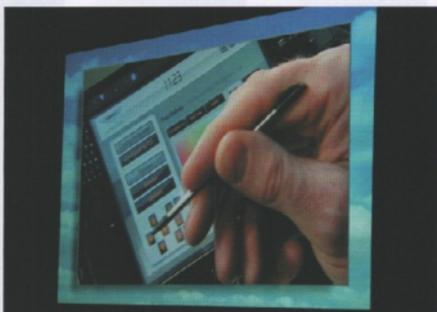
Mix est la grand-messe annuelle de Microsoft destinée aux acteurs du web 2.0 : graphistes, développeurs, etc. Sa troisième édition s'est tenue du 5 au 7 mars dernier à Las Vegas. L'occasion pour Microsoft de faire un point sur sa stratégie "software + service" et de présenter toutes les nouveautés.

" Notre façon de développer des applications va radicalement évoluer dans les 5 ans à venir " a prévenu Ray Ozzie, architecte en chef de Microsoft. La stratégie " software + services " va selon lui s'imposer car " le web devient un hub qui concentre les données personnelles et les interactions sociales, (...) interactions qui seront intégrées dans tous les logiciels sous la forme de services en ligne " explique-t-il.

Pour l'éditeur, les applications composites s'appuyant sur des ressources web et locales seront la règle d'ici quelques années. Microsoft entend donc fournir des outils qui facilitent la vie du développeur. Il a retenu trois grands axes : pousser son socle d'exécution universel .NET sur un maximum de systèmes d'exploitation et de périphériques, proposer des outils bien adaptés aux besoins des développeurs du web 2.0, et mettre à leur disposition de plus en plus de services en ligne.

Silverlight 2.0 : .NET et les mobiles

On savait déjà depuis un an que Silverlight 2 (ex Silverlight 1.1) intégrait une machine virtuelle .NET (CLR) et un " CLR dynamique " (DLR) pour les langages dynamiques tels que Ruby et Python. La bêta 1 intègre un sous-ensemble de l'API graphique WPF : WPF UI Framework. Il propose de nombreux objets d'interface, le développeur pourra concevoir de véritables winforms pour le web ! Dans les prochains mois, le framework s'enrichit encore (avant la sortie finale). Et en Open Source ! Les données peuvent être manipulées via LINQ (Language Inte-



Silverlight 2 et la mobilité.

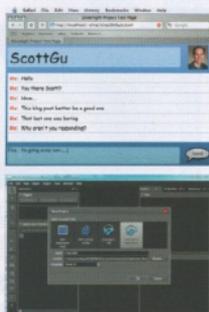
grated Query) et les échanges entre le client Silverlight et les services distants s'appuient sur différentes architectures : REST, SOAP, RSS / Atom Pub, etc. Au déploiement, Silverlight 2 prend la forme d'un exécutable de 4,3 Mo (sous Windows). Silverlight investit également les périphériques mobiles. Il s'exécutera d'ici la fin de l'année au-dessus de Symbian OS (Nokia) et Windows Mobile 6. " Nous sommes prêts à aller sur l'iPhone dès qu'Apple en exprimera le besoin et que les conditions commerciales s'y prêteront " précise Marc Jalabert, Directeur de la division Plateforme et Ecosystèmes chez Microsoft.

Silverlight 2.0 apportera de nouveaux scénarios d'utilisation. Une des nouvelles versions les plus spectaculaires est la possibilité d'adapter automatiquement le streaming vidéo selon le flux disponible chez le client, cela apportera une grande souplesse d'utilisation. Comme Adobe, Microsoft souhaite que les sociétés et développeurs puissent monétiser les sites et contenus

Silverlight. La version 2 permettra de mieux intégrer de la publicité en ligne, par exemple, grâce au streaming progressif, il est possible d'avoir une partie de la séquence gratuite et une autre payante. Cette version supporte bien IIS 7 pour le streaming et s'intègre avec Visual Studio 2008. La version pour Linux de Silverlight est en développement par l'équipe de Miguel de Icaza (auteur de Mono) au sein de Novell. De quoi élargir radicalement la portabilité des applications .NET !

Expression Studio 2

Très attendue, la suite Expression Studio 2 est également disponible en bêta. " Expression est une suite complète destinée aux designers Silverlight " précise David Talaga, chef de produit Expression chez Microsoft. " Cette deuxième mouture d'Expression Studio s'appuie sur WPF et Silverlight et répond aux besoins des web designers et web développeurs avec Expression Web, des intégrateurs WPF et Silverlight, des designers



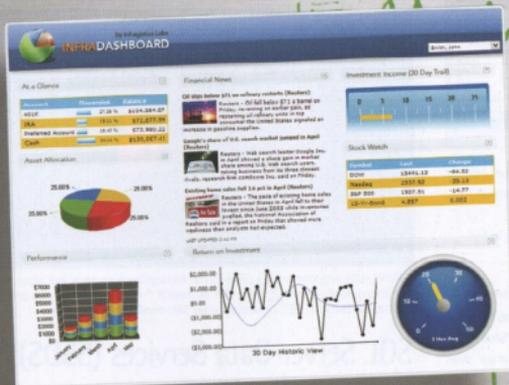
d'interface avec Expression Blend, et des designers graphiques avec Expression Design " précise-t-il. La liaison entre chaque module est assurée par un import-export XAML. Cette deuxième mouture sera déterminante pour l'adoption de Silverlight. Respectant mieux les standards, la suite intègre ASP.NET AJAX et Silverlight. Elle prend également en charge PHP (debug local via Cassini) et propose un support plus fin des calques Photoshop. " Expression Web 2 est un concurrent très sérieux à Dreamweaver " estime Yannick Lejeune, directeur de la stratégie de Ionis Education Group. Expression Blend 2.5 March Preview : Cette nouvelle version d'Expression Blend se distingue essentiellement de la version 2 par le support de Silverlight 2.

Internet Explorer 8

Outre de meilleurs outils, Microsoft mise aussi, et c'est assez nouveau pour le signaler, sur l'ouverture. En témoigne la nouvelle mouture d'IE 8 présentée pour la



NetAdvantage® for ASP.NET



NetAdvantage pour ASP.NET



Améliorez la performance de vos applications web Pour des applications Web 2.0 rapides

Des composants interfaces utilisateur de haute performance
Utilisez une boîte à outils complète pour ASP.NET propulsant vos applications commerciales demandant une performance rapide et de qualité

Infrastructure des normes d'industrie

Des XHTML et CSS soignés permettant des applications AJAX pour Web 2.0, ainsi que pour les WebParts de SharePoint®, réagissant correctement et faciles d'accès grâce à nos composants ASP.NET inclus dans le framework Aikido™

Compatibilité Visual Studio®

Utilisez la version de Visual Studio qui vous convient – VS 2005 ou bien VS 2008 avec un support complet et continu pour NetAdvantage

Concevez des tableaux de bord professionnels

Donnez de la vie à vos Key Performance Indicators avec des graphiques et gauges de haute fidélité pour ASP.NET, renforcés par des grilles splendides

Design Once, Style Everywhere

Donnez un style professionnel instantanément à vos applications conçues avec NetAdvantage afin de leur donner un style Office® 2007, XP, Vista® ou bien créez et appliquez vos normes corporatives à vos composants et à vos applications

Critères d'accessibilité du web

Proposez facilement des applications web accessibles pour tous, conformément aux normes des règles sur l'Accessibilité.



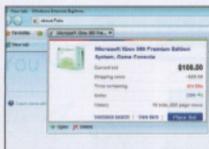
Pour de plus amples informations:
infragistics.com/dotnet

Infragistics
Paving the Presentation Layer

N°Vert 0800 667 307
sales-europe@infragistics.com

Copyright © 2008 Infragistics, Inc. All rights reserved. Infragistics, the Infragistics logo and Paving the Presentation Layer are registered trademarks of Infragistics, Inc. All other trademarks are the property of their owners.

première fois publiquement lors du salon. Cette version majeure supportera HTML 5 (support partiel, nldr). " Pour simplifier la vie des développeurs, nous avons intégré deux moteurs " explique Marc Jalabert. Si Ajax a pu apporter une nouvelle expérience, l'ergonomie n'est pas toujours au rendez-vous, par exemple la gestion exécutable du bouton retour. Avec HTML 5, on pourra enfin avoir une gestion fine de ce bouton dans le contenu dynamique... Internet Explorer 8 intègre deux moteurs de rendu. Le premier moteur est celui d'IE6 et IE7. Le second est un nouveau moteur qui " respecte mieux les standards du web, notamment leurs dernières moutures telles que CSS 2.1, HTML 5 et XHTML 1.1 " explique Scott Guthrie, vice-président de la division .NET Developer chez Microsoft. Internet Explorer 8 choisira automatiquement le moteur à utiliser en fonction des pages à afficher. Internet Explorer 8 intègre deux nouvelles fonctionnalités très intéressantes : WebSlices et Activités. WebSlices permet de délimiter un bloc d'une page à l'aide d'une annotation (un microformat basé sur Atom dont la spec est publiée sous licence Creative Commons). L'utilisateur peut alors bookmarker ce bloc. En survolant le bookmark, il obtient une mise à jour de l'information qui l'intéresse : cours de bourse, enchère, etc. Basée aussi sur un microformat public, Activités permet d'associer des actions à un type de contenu donné. Par exemple, afficher une carte en fonction d'une adresse. Activités propose exactement le même type de comportement que la gestion des microformats dans Firefox via Operator. Le standard demeure le maître mot d'IE8 et notamment sur un grand respect de CSS grâce à CSS Certification. D'autre part, IE8 bénéficie de tous les nouveaux outils de développement intégrés comme un



debug performance et un éditeur de code source assez évolué ! Pour en savoir plus : <http://blogs.msdn.com/ie/> <http://ie8.ebay.com/>

Softwares + services

L'ouverture du " nouveau " Microsoft se matérialise surtout dans les architectures d'échange avec ses services professionnels (Online) et grand public (Live) qui adoptent tous par défaut une interface REST. L'éditeur s'appuie pour cela sur ADO.NET Data Services (Astoria) qui expose et consomme n'importe quelle source de données via une interface REST et des échanges basés sur RSS / Atom Pub.

" Notre stratégie software + service consiste à héberger les mêmes services et API à trois niveaux différents : dans l'entreprise, chez des partenaires, au sein du "nuage" Microsoft " explique Marc Jalabert. L'éditeur distingue deux niveaux de services : ceux de " cloud Infrastructure " et ceux de " live platform ". Les services fonctionnels de la plate-forme Live (<https://account.live.com/services.aspx>) s'appuient sur ceux de plus bas niveau " cloud infrastructure " (<http://dev.live.com/>).

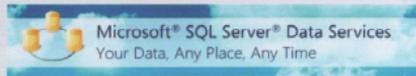
A l'image de SQL Server Data Service (SSDS), Microsoft met actuellement le paquet sur ses API de bas niveau pour proposer rapidement l'équivalent d'un framework complet en ligne. Microsoft préferait aussi un framework client qui simplifierait l'accès aux principales API du nuage. Affaire à suivre donc.

■ Carole Buret



" Nous vous proposons une plate-forme homogène pour tous vos types de développement : desktop, web, RIA et mobile " - Scott Guthrie, vice-président de la division .NET Developer chez Microsoft, lors du keynote.

SQL Server Data Services (SSDS)



For example, a query addressed to a container to retrieve all entities in that container having a "City" property equal to "Seattle" and a "State" property equal to "WA" would be written as follows:

```
from e in entities
where e["City"] == "Seattle" &&
e["State"] == "WA"
select e
```

In this query "entities" binds to the target container for the query.

Microsoft embrasse la notion de " cloud computing " et construit progressivement sa plate-forme " à la demande ". Il propose déjà Windows Live Storage (équivalent de S3 d'Amazon) et vient de lancer SQL Server Data Services (équivalent de SimpleDB). SQL Server Data Services (SSDS) sortira de sa phase de test début 2009. Il est déjà accessible en bêta. Cette base de données hautement disponible repose sur une infrastructure SQL Server. Microsoft s'engage sur les temps de réponse et la disponibilité via des SLA. Comme pour SimpleDB d'Amazon, l'accès et la manipulation des données s'effectue au choix via une interface REST ou SOAP et des messages XML. Mais il sera aussi possible d'utiliser le langage de requête LINQ depuis un client C# ou VB.NET car SSDS repose sur la technologie ADO.NET Data Services (Astoria). SSDS vise l'informatique à la demande, chère à IBM avec haute disponibilité incluant un stockage de données à la demande et des requêtes via des web services. On paie le stockage réel ainsi que toutes les ressources utilisées. La partie développement n'est pas oubliée avec des interfaces de programmation de type web services, Linq pour les requêtes, l'utilisation du modèle Entity. C'est un changement important dans l'approche de la base de données et cela confirme la volonté de Microsoft de proposer des services sur le principe du S+S ou SaaS selon le terme que l'on utilise. Reste maintenant à intégrer l'ensemble des services et à trouver une cohérence à tout cela ! On peut le voir comme une extension de SQL Server.

Développer sa première application Silverlight 2

Silverlight 2 a été annoncé le 5 Mars dernier. Il semble très prometteur. Nous allons donc nous pencher sur cette première bêta pour réaliser une première application Silverlight 2.

La première chose à faire, avant de commencer le développement, est de créer le projet. Pour cela, vous pouvez télécharger des outils pour Visual Studio 2008 qui vous permettront de le faire simplement (Fig.1).

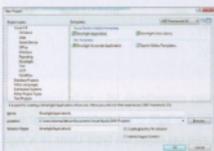


Fig.1

Une fois la demande de projet effectuée, Visual Studio vous affiche une boîte de dialogue vous expliquant que pour fonctionner, une application Silverlight doit être hébergée dans une page Web. L'assistant vous propose donc de créer une nouvelle application Web (ou un nouveau site) Web ou bien tout simplement une page HTML vous permettant de tester votre application (ce sera donc à vous de déployer l'application). Nous allons d'abord nous intéresser à comment héberger son application Silverlight (bien qu'ici, ce soit déjà fait). Dans le cas d'un développement ASP.NET, sachez qu'un nouveau tag fait son apparition : `<asp:Silverlight />`, le préfixe asp étant déclaré via la ligne suivante :

```
<%@ Register Assembly="System.Web.Silverlight" Namespace="System.Web.UI.SilverlightControls" TagPrefix="asp" %>
```

Ce tag prend plusieurs paramètres, le plus important étant source (qui permet d'indiquer le nom du fichier .xap à exécuter). Si vous ne souhaitez pas utiliser ASP.NET mais autre chose (HTML, PHP, etc.), vous devrez écrire ceci :

```
<object data="data:application/x-silverlight," type="application/x-silverlight-2" width="100%" height="100%">  
  <param name="source" value="ClientBin/TestConstellation.xap" />  
  <param name="onerror" value="onSilverlightError" />  
  <param name="background" value="white" />  
  
  <a href="http://go.microsoft.com/fwlink/?LinkId=108182" style="text-decoration: none;">  
      
  </a>  
</object>
```

Il s'agit d'un petit bout de code permettant d'initialiser l'application Silverlight, ou bien qui vous propose de télécharger le plug-in si celui-ci n'est pas installé.

En ce qui concerne l'application Silverlight, elle est composée (au départ) de 2 fichiers :

- App.xaml (et le code behind associé) : Fichier dans lequel l'on va pouvoir mettre les ressources de notre application
- Page.xaml (et le code behind associé) : Fichier qui correspond à la page principale de l'application Silverlight

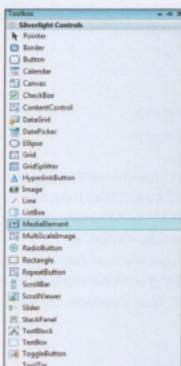


Fig.2

Votre première application

Nous pouvons donc à présent commencer le développement. Un coup d'œil à la boîte à outils nous montre que de nombreux contrôles sont disponibles (Fig.2). Si vous regardez la page principale, vous remarquerez que le contrôle n'est plus un Canvas (comme cela était le cas avec Silverlight 1.1) mais un UserControl. Nous allons donc commencer par créer un écran qui simulera une connexion à un site. Nous devons insérer des labels, des boutons et des zones de saisie. La première chose qui saute aux yeux, et qui vous fera gagner beaucoup de temps, est l'IntelliSense proposée par Visual Studio (Fig.3). Voici donc le code de notre application :

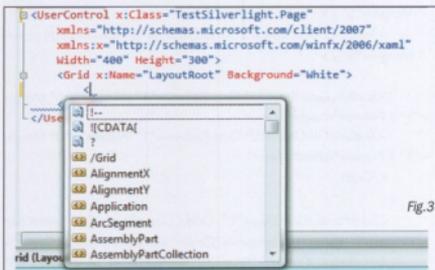


Fig.3

```
<UserControl x:Class="TestSilverlight.Page"  
  xmlns="http://schemas.microsoft.com/client/2007"  
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  
  Width="800" Height="Auto" xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006" mc:Ignorable="d"  
  xmlns:local="clr-namespace:TestSilverlight">  
  <Grid x:Name="LayoutRoot" Background="Gray" HorizontalAlignment="Center">  
    <Grid.ColumnDefinitions>  
      <ColumnDefinition />  
      <ColumnDefinition />  
    </Grid.ColumnDefinitions>  
  
    <Grid.RowDefinitions>  
      <RowDefinition Height="600" />  
    </Grid.RowDefinitions>
```

Événements

```
<RowDefinition Height="200" />
</Grid.RowDefinitions>

<Grid HorizontalAlignment="Stretch" Margin="7.9910001754
7607,7.99300003051758,8.00899982452393,8.006999969482
42" VerticalAlignment="Stretch" Grid.Row="0" Grid.ColumnSpan="2">
<Grid.RowDefinitions>
<RowDefinition Height="Auto"/>
<RowDefinition Height="Auto"/>
<RowDefinition Height="Auto"/>
</Grid.RowDefinitions>
<Grid.ColumnDefinitions>
<ColumnDefinition Width="50**"/>
<ColumnDefinition Width="Auto"/>
</Grid.ColumnDefinitions>
<TextBlock Grid.Row="0" Grid.ColumnSpan="2" Height="4
4.681" Margin="267.691986083984,35.6240005493164,189.24
6002197266,0" VerticalAlignment="Stretch" TextWrapping="Wrap"
HorizontalAlignment="Stretch" d:LayoutOverrides="Width">
<Run FontFamily="Portable User Interface" FontSize="14
666666984558106" FontStretch="Normal" FontStyle="Normal" Font
Weights="Normal" Foreground="#FF000000" Text="Bienvenue - Connectez
-vous !"/>
</TextBlock>

<TextBlock Grid.Row="1" Grid.Column="0" Text="Nom d'utilisateur:
" Margin="5" />
<TextBlock Grid.Row="2" Grid.Column="0" Text="Mot de passe:
" Margin="5" />

<TextBox Grid.Row="1" Grid.Column="1" Width="200" Margin
="5" x:Name="tbLogin" />
<TextBox Grid.Row="2" Grid.Column="1" Width="200" Margin
="5" x:Name="tbPwd" />
</Grid>

<StackPanel Grid.Row="1" Grid.ColumnSpan="2" Orientation
="Horizontal" HorizontalAlignment="Center" VerticalAlignment="Center">
<Button Content="Connexion" Width="Auto" Height="25"
Margin="5" x:Name="btConnect" Click="btConnect_Click" />
<Button Content="Effacer les champs" Width="Auto" Height
="25" Margin="5" x:Name="btClear" Click="btClear_Click" />
<Button Content="Récupérer son mot de passe" Width="Auto"
Height="25" Margin="5" />
</StackPanel>
</Grid>
</UserControl>
```

Dans le code behind, il ne reste plus qu'à écrire le code des méthodes qui seront appelées lorsque l'on cliquera sur les boutons :

```
private void btConnect_Click(object sender, RoutedEventArgs e)
{
    if (this.tbLogin.Text == "Thomas Lebrun" && this.tbPwd.Text ==
"pass")
    {
```

```
        User usr = new User() { Name = this.tbLogin.Text };

        MainForm form = new MainForm();
        form.DataContext = usr;

        this.Content = form;
    }

    private void btClear_Click(object sender, RoutedEventArgs e)
    {
        this.tbLogin.Text = string.Empty;
        this.tbPwd.Text = string.Empty;
    }
```

Comme vous pouvez le voir, rien de bien compliqué ici : si le login et le mot de passe correspondent (bien sûr, il s'agit d'un exemple à ne surtout pas reproduire en vrai), alors on crée un objet de type User (il s'agit d'un objet métier qui ne contient qu'une propriété Name), on instancie un nouveau UserControl auquel on définit notre objet User comme DataContext. Pourquoi faire cela ? Et bien tout simplement parce que sur notre objet MainForm, nous allons pouvoir faire un peu de DataBinding :

```
<UserControl x:Class="TestSilverlight.MainForm"
xmlns="http://schemas.microsoft.com/client/2007"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Width="400" Height="300">
<Grid x:Name="LayoutRoot" Background="White">
<TextBlock Text="{Binding Name}" />
</Grid>
</UserControl>
```

Personnaliser l'application

L'application que vous venez de développer utilise les styles et templates par défaut de Silverlight. Vous avez cependant la possibilité de les modifier afin de personnaliser, au mieux, votre application. Pour cela, il vous suffit de déclarer, dans les ressources de votre application ou de vos contrôles, les différents styles et templates à utiliser :

```
<Style TargetType="TextBox" x:Key="CustomTextBoxStyle">
<Setter Property="Margin" Value="5" />
<Setter Property="BorderThickness" Value="5" />
<Setter Property="BorderBrush" Value="Red" />
</Style>
```

Ensuite, il vous faut appliquer ce style aux textbox que vous voulez. Pour cela, rien de plus simple : il vous faut passer par la propriété Style du contrôle.

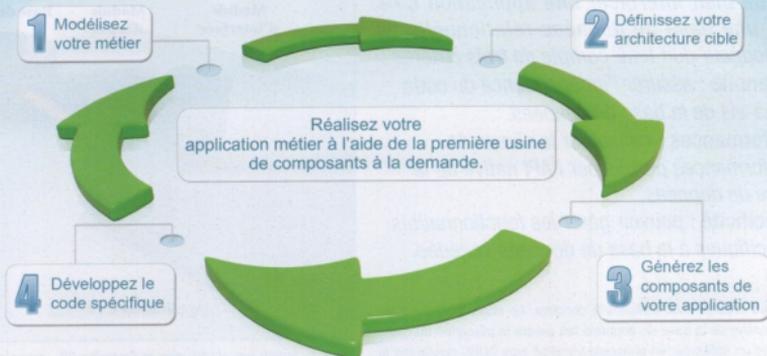
Et le style est appliqué automatiquement !

Bien sûr, il est possible de modifier (et de faire) beaucoup plus de choses mais cela nécessiterait un article plus complet que ce tour d'horizon.

Conclusion

Silverlight 2 est des plus prometteurs : un nombre impressionnant de contrôles, la possibilité de faire du DataBinding, d'appliquer des styles, des templates, etc. Le développement est très similaire au développement avec WPF : enfin un vrai Framework de RIA à la WPf en mode Web !

■ **Thomas Lebrun** - Consultant/Formateur - Winwise - Microsoft MVP C#
thomas.lebrun@winwise.fr - <http://blogs.developpeur.org/tom>



CodeFluent.com

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



NOUVEAU !

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

0€

CodeFluent

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



Prix public : ~~2490€ht~~

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

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



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

Softfluent recrute !

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



Développer une application C++ indépendante de la Base de Données

Afin de bien interfacer une application C++ avec une base de données relationnelle, le développeur doit tenir compte de trois critères :

- **Pérennité** : assurer l'indépendance du code vis-à-vis de la base de données.
- **Performances** : conserver le niveau de performances permis par l'API native de la base de données.
- **Spécificité** : pouvoir gérer les fonctionnalités spécifiques à la base de données accédée.

Ces critères sont difficiles à concilier. Le choix d'utiliser l'API native de la base de données fait perdre la pérennité du code, et un accès par un protocole banalisé type ODBC nivelle par le bas le niveau de performances et l'accès aux spécificités d'une base. Rogue Wave Software propose, avec la bibliothèque C++ SourcePro DB, une implémentation intéressante, qui satisfait aux trois critères, garantissant la pérennité sans sacrifier les performances et la spécificité. Autre avantage, cette bibliothèque est disponible sur de nombreuses plates-formes Windows, Unix et Linux, offrant ainsi une indépendance maximale du code par rapport à l'infrastructure.

Architecture de SourcePro DB C++

L'outil comporte un module d'interface qui présente au développeur l'API neutre d'accès aux bases de données. Ce module d'interface communique avec des modules d'accès, qui établissent une connectivité native à telle ou telle base de données. Chacun de ces modules d'accès interroge une base de données spécifique en invoquant son API native. On peut faire cohabiter plusieurs modules d'accès au sein de la même application. Cette architecture permet d'offrir une API commune, tout en s'appuyant sur les API natives fournies par les éditeurs de Bases, garantissant ainsi une conservation des performances.

Elle permet aussi de ne pas sacrifier les spécificités de chaque base de données. En effet, Rogue Wave a fait le choix de proposer une API riche en fonctionnalités, avec, pour chaque base de données, une émulation des fonctionnalités non gérées nativement par cette base. Rogue Wave fournit les modules d'accès pour toutes les bases majeures du marché, et les fait évoluer au fur et à mesure des versions. DB2, Informix, Microsoft SQL Server, MySQL Server, PostgreSQL Server, Oracle, Sybase, ainsi qu'un module banalisé ODBC, sont disponibles.

L'API de programmation

La solution présente au développeur, à travers le " Interface Module ", une API constituée de trois jeux de classes offrant un niveau plus ou moins élevé d'abstraction :

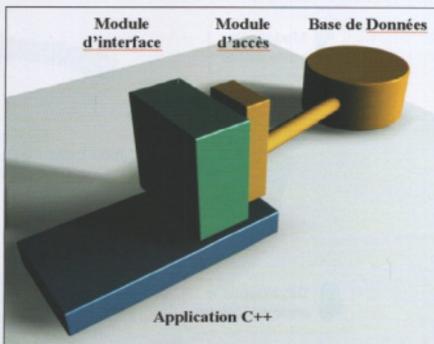


Figure 1 - Architecture et utilisation de SourcePro DB - exemple monobase

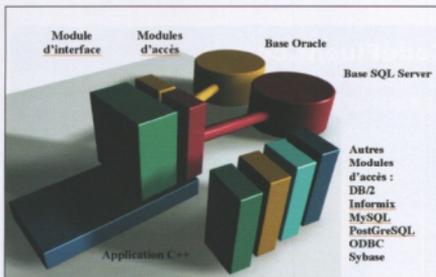


Figure 2 - Architecture et utilisation de SourcePro DB - exemple multi-bases

- Des classes C++ de haut niveau, offrant une abstraction maximale. Ces classes évitent de construire des requêtes SQL complètes, et gèrent complètement les accès à la base : pool de connexions, opérations sur les tables, jointures... Leur utilisation garantit la portabilité du code en cas de changement de Base de Données. Les développeurs utilisent généralement ces classes lors d'une implémentation.
- Des classes de plus bas niveau, nommées " Open SQL ". Assez similaires à JDBC, elles offrent un contrôle direct sur les requêtes SQL. L'utilisation de ces classes demande quelques précautions en programmation si l'on souhaite garantir une portabilité, le support de SQL étant différent d'un fournisseur de base de données à l'autre.
- Un accès direct à l'API native d'une base de données, à travers deux classes spécifiques implémentées dans le module d'accès corres-

pendant. Cet accès est volontairement proposé dans l'API Rogue Wave, pour laisser l'accès à des opérations très spécifiques d'une base. L'utilisation de cette API rend bien sûr le code spécifique de la base de données.

Une seule instruction pour changer de base

L'exemple illustre la facilité avec laquelle on change de base de données avec SourcePro DB. Il s'appuie sur une table MySQL comportant 3 colonnes : un ID de type Integer, un nom de type VarChar(30), une valeur numérique de type Float. Le programme lit le contenu d'une table MySQL et le déroule à l'écran. Examinons la ligne d'ouverture de la base de données. Elle comporte le nom d'une dll qui correspond au module d'accès MySQL, et des informations de connexion.

```
#include <iostream.h>
#include <rw/db/db.h>

int anInt;
RWCString aString;
float aFloat;

int main ()
{
    // Ouverture de la base de données MySQL
    RWDBDatabase aDB = RWDBManager::database("mys7012d.dll",
    "localhost", "guest", "guest", "test");

    RWDBSelector aSelector = aDB.selector(); // Sélecteur pour la base de données

    RWDBTable aTable = aDB.table("TABLE1"); //Choix de la table à accéder

    aSelector << aTable; // critère pour le sélecteur : toute la table TABLE1

    RWDBReader aReader = aSelector.reader(); // Balayage du résultat
    while (aReader())
    {
        aReader >> anInt >> aString >> aFloat; // lecture du reader
        std::cout << anInt << "\t" << aString << "\t" << aFloat << std::endl;
    }
    return 0;
}
```

Le développeur veut maintenant changer pour une base Oracle. Il suffit de modifier la ligne d'ouverture de la base de données, en mettant le nom de la dll qui correspond au module d'accès Oracle, et en changeant les informations de connexion.

Le reste de l'exemple fonctionne tel quel.

```
// Ouverture de la base de données Oracle
RWDBDatabase aDB = RWDBManager::database("ora7012d.dll",
"test", "scott", "tiger", "");
```

Notez aussi la simplicité du codage. Tous les concepts de base de données sont encapsulés par des classes simples d'gestion. Ces classes encapsulent également toute la gestion des allocations/

désallocations des connexions, statements, etc. qu'il faudrait gérer manuellement en ODBC.

Caractéristiques avancées

L'API ne se contente pas de gérer les connexions et les opérations basiques d'accès aux tables. Elle comporte des services supplémentaires, pensés pour offrir souplesse et performance.

Les opérations multi-tables sont bien sûr disponibles: support des jointures ANSI, des clés primaires et étrangères. En plus des requêtes traditionnelles, un jeu de classes permet de gérer intégralement la création et l'exécution des Procédures Stockées.

Les opérations en masse de "Bulk Copy" sont également proposées. A code identique, SourcePro DB invoque la fonctionnalité native à la base de données, ou un code d'émulation si la base ne sait pas gérer le Bulk en natif. Divers mécanismes permettant d'optimiser les performances sont disponibles : un cache de connexions, cache de schéma de la base côté client, cache d'échanges entre client et serveur.

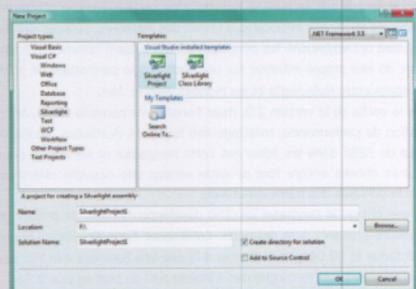
Toujours dans un souci de performance, des services permettent d'éviter le reparsing de requêtes similaires par le Query Optimizer de la base de données.

L'internationalisation n'a pas été oubliée, les "Character Sets" et locales sont pris en compte pour assurer un bon formatage des chaînes, dates et heures, nombres et unités monétaires.

Il est possible de gérer les accès à la base de manière asynchrone. Le choix du mode opératoire se fait lors de la connexion. Dans ce cas, une opération sur la base retourne un objet "Future" doté d'une méthode isReady(). Enfin, on dispose d'un support pour les transactions distribuées, avec le module SourcePro DB XA. Celui-ci permet de dialoguer avec un moniteur transactionnel compatible XA.

■ Jean Vidames

Visual 2008 Tools for Silverlight Bêta 1



Cet ad-on se greffe à Visual Studio 2008 auquel il ajoute la possibilité de créer un projet Silverlight 2, d'éditer le XAML correspondant, de déboguer, et de bénéficier du support d'IntelliSense (VD et C# pour le code behind). Il permet également à Visual studio 2008 de prendre en charge les projets Expression Blend. Cette bêta 1 comprend quatre contrôles (en CTP) : contact, IDLogin, IDLogin View, et SilverlightStreamingMedia Control. ■ C.B.

Ruby on Rails 2.0 : la nouvelle star du web !



Vous êtes de plus en plus nombreux à utiliser et développer avec Ruby et particulièrement avec le framework Rails. Depuis 2 ans, Rails s'est imposé comme une star du "web 2.0", à côté d'Ajax et des nouvelles applications internet riches. Programmez ! vous propose une plongée au cœur de Rails et tout particulièrement de Rails 2.0. Dans ce grand

dossier en deux parties, écrit et testé par les meilleurs spécialistes français du langage et du framework, nous aborderons les nouveautés de la v2, la sécurité, les tests, la migration, l'intégration avec Net-Beans, les aspects REST et l'intégration Ruby dans le framework .Net.

■ François Tonic

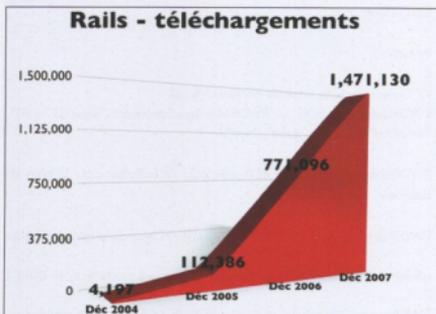
La maturité de Rails 2.0

À sa sortie en Décembre 2005, Rails 1.0 avait provoqué un véritable séisme dans le monde des frameworks de développement d'applications Web, une onde de choc qui a valu à Rails plusieurs distinctions dont le Jolt Award en 2006 et la "Hot Technology no.1" de ComputerWorld en 2007. S'appuyant sur des principes forts (pas de duplication de code, adoption de convention plutôt que de longs fichiers de configuration, adhésion stricte au modèle de programmation MVC, un seul langage – Ruby – pour tout faire) Rails balayait d'un revers de main 10 années de confusion et de cacophonie dans le domaine des frameworks de développement Web.

Avec plus de 700 000 téléchargements au cours des 6 derniers mois, Rails s'impose de plus en plus comme le framework Web 2.0 de référence et, par effet de bord, le langage Ruby gagne lui aussi en popularité en prenant récemment deux places au classement Tiobe [2]. Un tel engouement a mis la puce à l'oreille d'un certain nombre d'acteurs industriels du monde informatique et non des moindres, puisque Sun et Microsoft ont embauché les développeurs qui avaient commencé à travailler de leur propre initiative sur une technologie permettant de jeter des ponts entre Ruby/Rails et les mondes Java et .Net.

Avec la sortie de la version 2.0, Rails franchit une nouvelle étape. Amélioration de performance, toilettage des fonctions inutilisées, adoption totale de REST dans les échanges entre navigateur et serveur et bien d'autres choses encore font de cette version une nouvelle référence dans le domaine des frameworks Web.

Comme on pu le constater les 250 développeurs, chef de projets et décideurs présents à la deuxième conférence Paris On Rails [3] qui s'est tenue le 10 Décembre dernier à la cité des Sciences à la Villette, les entreprises petites ou grandes s'intéressent de plus en plus à Rails. La banque RBC Dexia IS, deuxième banque mondiale de services aux investisseurs, est ainsi venue témoigner des gains de productivité spectaculaires réalisés depuis qu'ils ont fait le choix de développer leurs applications départementales internes avec Rails plutôt qu'avec Java. La cohabitation entre les deux frameworks se fait sans douleur : Java continue de régner en maître incontesté (et à juste titre) sur la partie backoffice des processus de l'entreprise (data warehouse, transactionnel, etc.) et Rails apporte son agilité et son expressivité pour les applications manipulées par l'utilisateur final.



Moins c'est mieux

S'il est une loi quasi universelle dans le monde de l'informatique c'est bien celle qui impose à un éditeur de logiciel d'en rajouter toujours plus dans ses produits. Résultat : 80% des développeurs finissent par n'utiliser que 20% des fonctionnalités.

Rails est à l'opposé de cette spirale infernale. La 2.0 apporte certes son lot de nouveautés mais son créateur, David Heinemeier Hansson est tout aussi fier, et à juste titre, d'avoir su enlever un certain nombre de fonctionnalités qui n'ont pas résisté à l'usage. Elles ont été soit supprimées purement et simplement, soit extraites du framework pour devenir des plug-in optionnels. Parmi les exemples les plus marquants, on peut citer les primitives `acts_as_*` (tels que `acts_as_list`, `acts_as_tree`, etc.) qui permettent de manipuler les objets d'un modèle comme les éléments d'une liste ou comme les nœuds d'un arbre. Ces primitives qui influent sur le comportement d'un modèle sont certes très utiles mais elles ne font pas partie à proprement parler du cœur de Rails et sont donc devenues des plug-in externes. D'autres d'autres plug-in `acts_as_*` sont apparus depuis et viennent compléter la collection (`acts_as_versioned` – archivage automatique des enregistrements modifiés, `acts_as_paranoïd` – aucun enregistrement de la base n'est



détruit, etc.). La primitive `paginate` qui permet de fragmenter une liste de résultats en pages successives a subi le même sort: cette fonction très utile a été améliorée et transformée en `will_paginate`. Les pilotes de bases de données ont eux aussi subi une cure d'amaigrissement. Dans sa distribution standard 2.0, Rails offre uniquement les adaptateurs pour les bases de données Open Source à savoir SQLite, MySQL et PostgreSQL. Les autres bases de données propriétaires sont installables séparément sous forme de gems (paquetages) Ruby dont le nom est de la forme `activerecord-xxx-adapter` où `xxx` représente le nom de la base de données (Oracle, db2, etc.). Au chapitre des disparus on trouve les 'composants', une belle idée sur le papier consistant à regrouper HTML et code applicatif sous forme de composants réutilisables, mais qui n'a pas convaincu. L'idée n'a pas été abandonnée mais l'implémentation n'était pas convaincante, les 'composants' qui avaient déjà été dépréciés dans la version 1.2 sortent donc définitivement de la 2.0. ActionWebService, la couche de services web XML/RPC et SOAP a elle aussi été extraite du cœur du framework et doit maintenant être installée séparément.

Améliorations sur les performances

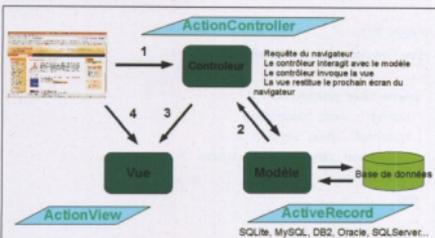
On entend souvent tout et n'importe quoi sur les performances du langage Ruby et donc de Rails, certains prenant argument du fait que Ruby est intrinsèquement moins rapide que C/C++ ou Java, pour dénigrer le framework. Pour reprendre une phrase célèbre "c'est un peu court jeune homme". On peut bien évidemment toujours gagner en performance pure mais tels qu'ils sont aujourd'hui, Ruby et Rails propulsent de nombreuses applications Web qui traitent allègrement des centaines de milliers voir des millions d'accès par jour. La version 2.0 apporte néanmoins son lot d'améliorations dans ce domaine :

- cache des requêtes (query caching) : en complément des mécanismes de cache de fragment HTML et d'actions complètes d'un contrôleur, Rails dispose maintenant d'un cache au niveau des requêtes SQL. Si deux requêtes identiques parviennent à ActiveRecord (la couche Modèle du framework) sans qu'aucune opération de mise à jour de la base n'ait eu lieu entre temps, Rails n'effectuera cette requête qu'une seule fois.
- Garnitures de tests (fixtures) : les jeux de données utilisés dans les tests (fixtures) ont non seulement été améliorés au niveau de la syntaxe (voir foxy fixtures ci-dessous) mais leur traitement est beaucoup plus performant, ce qui rend les tests de 50 à 100% plus rapides.
- asset server and caching : pour des raisons historiques (liées à la faiblesse de la bande passante réseau à l'époque des modems RTC) les navigateurs modernes s'imposent de ne pas ouvrir plus de deux connexions en direction d'un même serveur Web. Pour contourner cette limitation, Rails vous permet de déclarer un même serveur Web sous plusieurs noms (comme `www1.zya.com`, `www2.zya.com`, etc.) et utilisera ces différents noms à tour de rôle pour donner l'impression au navigateur qu'il communique avec N serveurs. Ainsi le navigateur ouvre davantage de connexions et charge plus rapidement la page à visualiser.
- gestion des sessions par cookies: après des discussions très nourries, l'équipe de développement de Rails a pris la décision de stocker les données afférentes à une session utilisateur dans un cookie. C'est le mode par défaut utilisé par Rails car c'est de loin le plus rapide et celui qui nécessite le moins d'entretien côté serveur. Les autres modes de stockage (sur le système de fichier, dans la base de données, en mémoire partagée ou en mode distribué) sont toujours disponibles. D'autre part Rails met en place un mécanisme de signature

des données de session stockées dans le cookie (digest HMAC SSL) ce qui empêche les utilisateurs mal intentionnés de forger des valeurs de session à la main et d'abuser votre application.

Nouveautés sur les modèles (ActiveRecord)

ActiveRecord est l'ORM (Object-Relational Mapper) de Rails, c'est-à-dire le module responsable de la mise en correspondance d'une table de votre base de données avec une classe Ruby qui hérite de la super classe ActiveRecord::Base. ActiveRecord permet de manipuler les enregistrements de vos tables comme des objets, et les colonnes des tables comme les attributs d'un objet.



ActiveRecord était déjà réputé pour sa puissance et sa grande simplicité d'usage mais il a encore été amélioré sur plusieurs points. Les validateurs, ces méthodes qui contrôlent que les objets entrants et sortants de la base de données répondent à certains critères comme la valeur non vide d'un attribut, le fait qu'il soit numérique, etc. acceptent des options supplémentaires

```
class MonEntite < ActiveRecord::Base
```

```
# Vérifie que la valeur de l'attribut 'prix' est numérique
# et supérieure à 1 centime.
validates_numericality_of :prix, :greater_than => 0.01
```

```
# Vérifie que la valeur de l'attribut 'nombre premier'
# est bien impair
validates_numericality_of :nombre_premier, :odd => true
```

```
# L'attribut 'note' doit être numérique mais la valeur vide
# est permise
validates_numericality_of :note, :allow_blank
```

```
end
```

Les migrations, des fichiers écrits en Ruby permettant de créer, supprimer ou modifier les tables utilisées dans votre base de données qu'il s'écrit désormais de manière plus condensée. Il est possible de déclarer plusieurs colonnes du même type sur une seule ligne, les colonnes de type marqueurs de temps (`created_at` et `updated_at`) peuvent s'abrégier en `timestamps` et les références à d'autres tables se déclarent comme telles. Voici un exemple qui construit une table de produits :

```
# Rails 1.2
```

```
class CreateProduits < ActiveRecord
```



```
...
create_table :produits do |t|
  t.column :string, :nom
  t.column :string, :description
  t.column :decimal, :prix, :scale => 2
  t.column :integer, :category_id, :null => false
  t.column :datetime, :created_at
  t.column :datetime, :updated_at
end
...
end
```

```
# Rails 2.0
class CreateProduits < ActiveRecord
  ...
  create_table :produits do |t|
    t.string :nom, :description
    t.decimal :prix, :scale => 2
    t.references :category, :null => false
    t.timestamps
  end
  ...
end
```

Et pour terminer sur le chapitre ActiveRecord, nous parlerons des améliorations apportées à la syntaxe des fichiers de jeux de données (fixtures) chargés préalablement à l'exécution des tests. Supposons que nous ayons deux classes Produit et Catégorie liées par une relation 1:N (une catégorie peut comporter N produits et 1 produit appartient à une seule catégorie).

```
class Produit < ActiveRecord::Base
  belongs_to :category
end

class Catégorie < ActiveRecord::Base
  has_many products
end
```

Dans cette situation il est naturel de créer un jeu de données par classe. En Rails 1.2 il fallait que le jeu de données déclare explicitement l'ensemble des colonnes et leur valeur pour chaque enregistrement à créer dans la base de données de test. En 2.0 il n'est plus nécessaire de déclarer explicitement la valeur de la clé primaire (il s'agit désormais d'un hachage créé à partir du nom de l'enregistrement), ni les champs marqueurs de temps (qui seront automatiquement initialisés à la valeur de temps présente Time.now). D'autre part il n'est plus nécessaire d'utiliser les id des enregistrements pour exprimer les relations qui lient les enregistrements entre eux, il suffit d'utiliser leur nom. Voici un exemple d'ancienne syntaxe avec un enregistrement dans chaque table:

```
# fichier produits.yml
tshirt:
  id: 1
  nom: T-shirt
  description: T-shirt pur coton
  prix: 7.99
```

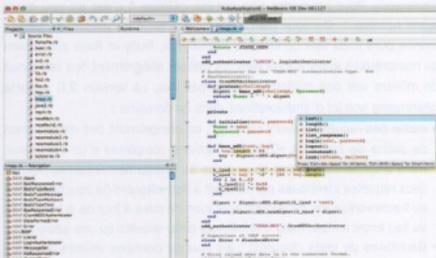
```
category_id: 1
created_at: <%= Time.now %>
updated_at: <%= Time.now %>
```

```
# fichier categories.yml
vetement:
  id: 1
  nom: Vêtement
```

Avec la nouvelle syntaxe les fixtures sont beaucoup plus ramassées, plus lisibles et surtout beaucoup plus maintenables grâce à l'utilisation des noms d'enregistrement dans les références:

```
# fichier produits.yml
tshirt:
  nom: T-shirt
  description: T-shirt pur coton
  prix: 7.99
  category: vetement # mettra category_id 1
```

```
# fichier categories.yml
vetement:
  nom: Vêtement
```



REST

Cette nouvelle version de Rails pousse encore plus loin son orientation REST et confirme, s'il en était besoin, qu'il s'agit de l'architecture de référence choisie par les concepteurs du framework pour le développement d'applications Web. La nomenclature initiale qui posait parfois des problèmes avec certains serveurs web a été modifiée et le point-virgule qui était utilisé comme séparateur a été remplacé par la barre oblique. Vous trouverez tous les détails concernant les nouveautés REST un peu plus loin dans ce dossier.

Les vues

Comme les autres composants de Rails 2.0, la couche Vue à elle aussi été améliorée afin de simplifier encore plus son utilisation. Désormais le format des vues est indépendant du moteur de rendu à utiliser, ainsi une vue précédemment nommée "index.rhtml" devient avec Rails 2.0 "index.html.erb" et sera le template par défaut qui sera utilisé pour une action "index". La syntaxe est donc maintenant "action.format.moteur_de_rendu", par exemple "create.js.js" pour

Extrême
Java

Seam

UML

Hibernate

valtech
training

Gestion
de projet

Scrum

XML

Au plus court vers
vos nouvelles compétences

Architecture et intégration

- Introduction au logiciel libre (1 jour)
- La persistance dans les applications Java (1 jour)
- Ingénierie logicielle objet (3 jours)
- Introduction aux architectures et technologies du Web (1 jour)
- Architectures .Net multi-niveaux (3 jours)
- Intégration d'applications (EAI, B2B) : les technologies et le projet (3 jours)
- Urbanisation du système d'information (2 jours)
- Architecture orientée service (3 jours)
- Architecture d'entreprise avec Java EE (4 jours)
- Du Mainframe au serveur d'applications (1 jour)

Développement Java et C++

- Introduction technique à Java (1 jour)
- Programmer en utilisant les aspects et les Design Patterns (3 jours)
- Java et la conception objet (5 jours)
- Développement d'un client riche avec SWT et Eclipse RCP (3 jours)
- Atelier Java avancé (5 jours)
- Eclipse, créer son environnement de développement intégré (2 jours)
- Programmation intensive avec Java (5 jours)
- Extrême Java (4 jours)
- Développer une application Corba (4 jours)
- L'essentiel de C++ et la conception objet (5 jours)
- Programmation efficace et avancée en C++ (5 jours)

Microsoft .Net

- C# et la conception objet (5 jours)
- Programmation avec Visual Basic .Net et conception objet (5 jours)
- Programmation intensive avec le Framework .Net (5 jours)
- Développement d'applications Web avec ASP.NET (5 jours)
- Développement d'applications Windows Forms sur la plate-forme .Net (5 jours)
- Développer des applications avec C# et le Framework .Net 3.0 (5 jours)

Gestion de projet

- Gérer des projets avec un processus itératif (4 jours)
- Les méthodes agiles de développement logiciel (1 jour)
- Le Processus Unifié de développement logiciel (2 jours)
- Du recueil des besoins aux exigences : rédiger le cahier des charges (2 jours)
- Gestion de projet (3 jours)
- Manager des hommes dans le cadre d'un projet (2 jours)
- Management de projet (5 jours)
- MSProject (3 jours)
- Gérer les projets agiles avec Scrum (2 jours)
- Gérer les projets agiles avec XP (2 jours)

.Net

Frameworks Java EE

- Concevoir et développer des EJB 2 (5 jours)
- Développer une application Java EE avec les EJB 3 (5 jours)
- Gestion de la persistance avec Hibernate (3 jours)
- Mise en œuvre du Framework Seam (3 jours)
- Développement avec le Framework Spring (3 jours)
- Gestion avancée de la persistance avec Hibernate (2 jours)

Analyse, conception et modélisation avec UML

- Introduction technique à l'analyse, la conception et la programmation objet (1 jour)
- Introduction à UML (1 jour)
- Concevoir avec les Design Patterns (3 jours)
- La modélisation métier avec UML (5 jours)
- Analyse et conception avec UML (5 jours)
- La modélisation des systèmes complexes avec UML 2 et SysML (3 jours)
- La modélisation efficace des exigences avec les cas d'utilisation (2 jours)
- Analyse orientée objet avec UML (2 jours)
- D'UML 1 à UML 2 : quoi de neuf, docteur ? (1 jour)
- Modéliser les besoins et analyser avec UML (4 jours)

Oracle

- Introduction technique (1 jour)
- Exploitation (4 jours)
- SQL (3 jours)
- PL / SQL (3 jours)
- Optimisation des requêtes (2 jours)
- Administration (5 jours)
- Tuning (3 jours)

XML et Web Services

- Introduction à la technologie XML (1 jour)
- Introduction aux technologies Web Services (1 jour)
- Développer avec XML (3 jours)
- Développer une application XSL (2 jours)
- Développer des applications Web Services en Java (3 jours)
- Développer des applications XML avec Java (2 jours)

90 formations
au développement logiciel

chez vous
ou à Paris, Toulouse, Lyon, Grenoble,
Genève, Bruxelles, Luxembourg

Développement Web

- Développement de pages Web avec HTML, CSS et JavaScript (3 jours)
- Développement, déploiement et administration d'applications Web (Java EE) avec WebSphere (3 jours)
- Développement d'applications Web avec PHP (3 jours)
- Ajax, pour dynamiser le poste client (2 jours)
- Hacking des applications Web (1 jour)
- Développer des applications avec Adobe Flex 3 (5 jours)
- Conception d'applications Web d'entreprise avec Java EE, les Servlets, JSP et Struts (5 jours)
- Développement d'applications Web avec Ruby on Rails (3 jours)
- Développement d'applications Web avec JavaServer Faces (3 jours)
- JavaServer Faces avancé (2 jours)
- Utilisation du Framework Struts pour le développement d'applications Web (3 jours)
- Développer une application Web avec Ajax et le Google Web Toolkit (3 jours)
- Développer des applications pour Adobe Integrated Runtime (2 jours)

Stratégies de développement logiciel

- Test Driven Requirement ou la gestion des exigences dirigée par les tests (2 jours)
- Test Driven Development ou la programmation pilotée par les tests en Java (3 jours)
- Stratégie de test, vérification et validation (3 jours)
- Les fondements de l'IT Infrastructure Library (ITIL) (3 jours)
- Introduction au CMMI (3 jours)
- L'usine logicielle, des concepts à la pratique (3 jours)



www.valtech-training.fr
+33 (0)1 41 88 23 00 - +33 (0)5 62 47 52 00
info@valtech-training.fr

utiliser un template RJS avec l'action " create ". Tous les helper sont maintenant inclus par défaut et accessibles par toutes les vues grâce à la commande " helper :all " qui se trouve dans le fichier application.rb. Un nouveau niveau de layout a été ajouté afin de permettre de factoriser encore plus finement les parties réutilisables des vues, il s'agit des layout partiels. Le schéma de nommage des partiels est conservé et le nom de ce fichier doit commencer par " _ " (tiret bas) et il doit se trouver par défaut dans le même répertoire que les vues qui vont l'utiliser. Vous pouvez donc désormais utiliser l'option :layout lors de vos appels de fichiers partiels :

```
<%= render :partial => ÂformÃ, :layout => ÂsimpleÃ, :locals => { :article => @article }
```

Notez aussi le fait que toute variable passée par l'option :locals est non seulement accessible au fichier partiel mais aussi au fichier de layout. Au niveau sécurité tous les formulaires sont maintenant protégés par défaut contre les attaques CSRF (Cross Site Request Forgery) grâce à la commande " protect_from_forgery " qui se trouve dans le fichier application.rb. Et dans le cas où vous auriez besoin de désactiver cette fonctionnalité globalement vous pouvez soit mettre en commentaire la ligne précédente soit définir " config.action_controller.allow_forgery_protection = false " dans le fichier environment.rb.

Goodies pour le développeur

Outre les modifications internes au framework, Rails 2.0 n'a pas oublié que le cycle de vie d'un projet doit s'appuyer sur des outils périphériques efficaces et un certain nombre d'améliorations ont été apportées dans ce domaine.

Tout d'abord l'outil rake (l'équivalent de make en C ou ant en Java) offre de nouvelles tâches prédéfinies qui facilitent la vie du développeur :

rake db:reset	Recrée la base et son jeu de données à partir de zéro en une seule commande.
rake db:rollback STEP=12	Permet de revenir à une version antérieure de la base de données (niveau de migration)
rake db:create:all	Crée toutes les bases de données (locales) définies dans config/database.yml
rake db:create or db:drop	Crée ou détruit la base de données
rake db:version	Affiche le numéro version courant de la base de données.

Rails 2.0 offre aussi au développeur la possibilité d'annoter son code par l'intermédiaire de commentaires comportant certains mots-clés: (TODO) pour les choses qui restent à faire, (FIXME) pour résoudre ultérieurement un problème identifié et (OPTIMIZE) pour signaler que le code doit être optimisé. Par la suite, la commande rake notes vous donnera la liste de toutes les annotation et rake notes:fixme, rake notes:todo ou rake notes:optimize la liste des annotations d'un certain type. Enfin, la nouvelle version de Rails a complètement remis à plat le debugger en abandonnant l'utilisation du module breakpoint qui ne donnait guère satisfaction au profit du gem (paquetage) de débogage rapide de Ruby baptisé ruby-debug.

Pour utiliser le nouveau debugger il faut d'abord installer ruby-debug avec la commande gem install ruby-debug puis insérer l'instruction debugger à l'endroit de votre code source où vous souhaitez rentrer en mode débogage. Voici quelques unes des commandes disponibles dans le debugger :

app.get 'list?limit=10'	Permet de solliciter une action de votre application, ici l'action list avec un paramètre
list	Liste de code de part et d'autre du point d'exécution courant
p variable	Inspecte une variable (locale, d'instance, de classe, globale...)
next	Passé à la ligne suivante sans rentrer dans un éventuel appel de méthode
step	Passé à la ligne suivante en entrant dans un éventuel appel de méthode
up	Quitter la méthode en cours pour retourner à l'appelant
Cont	Continuer l'exécution en quittant le debugger

Le futur

Cet article ne peut malheureusement pas décrire toutes les autres améliorations apportées à Rails 2.0. Vous pourrez trouver une liste plus détaillée sur le blog de DHH [4]. A n'en pas douter Rails va continuer à évoluer dans les mois qui viennent et il y a fort à parier qu'il gardera sa position de leader en matière d'innovation. Néanmoins, nous ne saurions trop conseiller à nos lecteurs de garder aussi un œil attentif sur le langage Ruby lui-même dont les nouveautés attendues pour 2008 devraient avoir un impact majeur sur l'expansion de Rails. En tête de ces nouveautés figurent les projets de machine virtuelle Ruby. Parmi les 4 projets en cours, deux retiennent particulièrement l'attention: YARV et Jruby. YARV est le projet officiel de machine virtuelle Ruby menée par Yukihiro Matsumoto (dit " Matz "), l'inventeur de Ruby. La machine YARV est déjà intégrée dans la version expérimentale 1.9 de Ruby et annonce déjà un facteur d'amélioration des performances d'un facteur 2 à 4 dont Rails bénéficiera directement. L'autre projet à suivre avec grande attention est le projet Jruby(5) de Sun. Avec Jruby, Sun va implanter pour la première fois de façon officielle un langage fortement dynamique au-dessus de la machine virtuelle Java (JVM). Exercice intellectuel intéressant me direz-vous mais à quel cela va-t-il bien pouvoir servir? Et bien tout simplement à jeter les ponts entre le monde de l'entreprise grand consommateur de programme Java et l'agilité de Rails pour le développement d'applications Web départementales. Jruby 1.0.2 n'est pour l'heure qu'un interpréteur Ruby mais il fonctionne déjà très bien et il permet de faire tourner Rails tout en ayant accès à l'immense collection de classes Java. Dans la version 1.1, déjà disponible en version bêta, Sun propose la compilation du code en mode Just-in-time. De nombreuses entreprises regardent Jruby comme une sorte de Saint Graal et il y a fort à parier que nous en reparlerons en 2008.

Références

- [1] Rails: <http://www.rubyonrails.org>
- [2] <http://www.tiobe.com/tpci.htm>
- [3] <http://paris.onrails.info>
- [4] <http://weblog.rubyonrails.com/2007/12/7/rails-2-0-it-s-done>
- [5] <http://jruby.codehaus.org/>



■ Laurent Julliard

Utilisateur de Ruby depuis 2000. Traducteur de l'ouvrage "Agile Web development with Rails" par Eryolles, il a en outre participé à plusieurs projets Ruby dont l'environnement de développement FreeRIDE. Directeur Associé de Nuxos Group.



■ Richard Piacentini

Initiateur et organisateur de la conférence "Paris on Rails", co-fondateur de la SSL Nuxos Group spécialisée dans les développements, créateur du portail Railsfrance.org et validateur technique de l'édition française de la bible des rails: "Ruby on Rails" (Ed Eryolles) et "Recettes Rails" (Ed Pearson).



Tout pour REST

REST est le "buzzword" qui fait probablement le plus fureur depuis quelque temps dans la communauté Rails. Et à raison : il s'agit ni plus ni moins d'une manière tout à fait élégante de structurer son application et de la rendre de facto accessible de manière unifiée, par les humains comme par les machines. Pourtant, REST n'est pas un concept nouveau, et n'est certainement pas spécifique à Rails.

REST (Representational State Transfer) est un style d'architecture logicielle pour systèmes distribués tels que le Web; le terme a été proposé par Roy Fielding – l'un des auteurs du protocole HTTP - dans sa thèse de doctorat en 2000. Ce style d'architecture est orienté autour du concept de "ressources" qui correspondent à l'ensemble des objets exposés par le système. On dit d'un système respectant les principes de REST qu'il est "RESTful". Un tel système utilise la simplicité du Web et se place à l'opposé d'autres systèmes tels que SOAP ou XML-RPC, adaptés au style RPC (Remote Procedure Call) bien plus lourd à mettre en oeuvre. Pour être RESTful, un système doit simplement se conformer aux principes suivants :

- **adressabilité** : il doit exposer son modèle de données sous la forme de ressources ; chacune de ces dernières doit posséder une URI propre qui permet d'y accéder ;
- **absence d'état** : chaque requête HTTP doit avoir lieu de manière totalement isolée ; concrètement, cela implique qu'aucun état propre au client (s'il est loggé, l'étape courante d'un processus d'enregistrement, etc.) ne peut être retenu par le serveur ; l'état propre à une ressource se trouve par contre bien entendu sur le serveur (dans la DB, dans un fichier, ...) ;
- **connectivité** : le fait d'avoir des liens permettant de naviguer d'une ressource à l'autre, et de modifier leur état ;
- **interface uniforme** : le nombre et la nature des actions que l'on effectue couramment sur une ressource se limite à 4, correspondant à 4 méthodes définies par HTTP :
 - Récupérer une représentation d'une ressource : HTTP GET
 - Créer une nouvelle ressource : HTTP PUT (vers une nouvelle URI) ou HTTP POST (vers une URI existante)
 - Modifier une ressource existante : HTTP PUT (vers une URI existante)
 - Supprimer une ressource existante : HTTP DELETE

REST versus SOAP

REST se présente donc comme une alternative à SOAP, dans le sens où il permet d'exposer votre application à d'autres systèmes non-humains via une interface de programmation (ou API). Le problème de SOAP, du moins lorsqu'on adopte le style RPC, ce qui est généralement le cas, est qu'il nécessite de définir cette API en explicitant le nom et les arguments des méthodes. Il n'y a aucune standardisation : plusieurs applications possèdent donc plusieurs API différentes et il devient très vite compliqué de s'y retrouver (il suffit de comparer les API SOAP de Flickr, Facebook, LinkedIn, etc. pour s'en convaincre).

Par opposition, avec REST, vous ne devez pas définir d'API, ou plutôt, vous la définissez implicitement en structurant votre application : les différentes ressources que vous utilisez ne proposent que les 4 méthodes de l'interface uniforme (GET, POST, PUT, DELETE). Ces quatre méthodes sont les mêmes, que l'utilisateur soit un humain ou un ordinateur.

REST dans Rails 2.0

Rails propose une architecture basée sur REST, qui comporte certaines particularités et limitations. Pour présenter cette architecture, nous allons prendre l'exemple classique du blog : une application constituée d'articles et de commentaires, pouvant être consultée en html dans un navigateur, mais également en xml ou rss dans un agrégateur de flux.

Routes

Dans une application classique, vous rajoutez dans config/routes.rb un ensemble de routes nommées; par exemple, les routes suivantes redirigent vers des méthodes spécifiques du contrôleur ArticlesController :

```
map.list_articles "articles/index", :controller => "articles"
map.show_article "articles/:id", :controller => "articles", :action => "show"
```

Ces chemins peuvent ensuite être appelés via les méthodes ainsi créées :

```
list_articles_path
show_article_path(:id=>24)
```

Dans un modèle REST, il n'est pas nécessaire de nommer chaque route individuellement. Avec la méthode "resource", toutes les routes REST correspondantes à la ressource sont automatiquement produites. Dans notre cas, pour rajouter la ressource "article" :

```
ActionController::Routing::Routes.draw do |map|
  map.resources :articles
end
```

Les 7 routes principales générées sont reprises dans le tableau ci-contre. On peut constater que l'usage des verbes HTTP permet de ne plus avoir à spécifier l'action dans l'URL (show, destroy, etc.) que chaque URL représente uniquement une ressource et c'est le verbe qui détermine ce que l'on en fait. Notons cependant les exceptions "new" et "edit" qui ne sont pas des actions REST à proprement parler, mais qui sont nécessaires pour qu'un utilisateur humain puisse ajouter/éditer une ressource dans un navigateur.

Méthode	Verbe HTTP	Chemin	Action
articles_path	GET	/articles	index
article_path(id)	GET	/articles/:id	show
new_article_path	GET	/articles/new	new
articles_path	POST	/articles	create
edit_article_path(id)	GET	/articles/:id/edit	edit
article_path(id)	PUT	/articles/:id	update
article_path(id)	DELETE	/articles/:id	destroy

De plus, diverses variations sont proposées :

- remplacer le suffixe "_path" par "_url" pour renvoyer le chemin com-



plet avec le protocole, l'hôte, et le port. Ex. "articles_url" => "http://localhost:3000/articles"

- ajoutez le préfixe "formatted_" pour indiquer le format souhaité. Ex. formatted_article_path(1, :format=>:xml) => /articles/1.xml

- ajoutez le préfixe "hash_for" pour renvoyer un hash et non une chaîne de caractères. Ex. hash_for_articles => { :only_path => true, :use_route => "articles", :controller => "articles", :action => "index" }

Ces variations sont combinables (ex.: hash_for_formatted_articles_url { :format => :rss }).

Pour chaque ressource définie dans routes.rb, Rails s'attend à ce que l'on crée un contrôleur disposant en tout et pour tout de 7 actions. Ces 7 actions permettent d'exposer à la fois l'objet lui-même (un article), et sa liste (la liste des articles présents dans le système) :

```
class ArticlesController < ApplicationController
  def index; end # /articles (GET)
  def new; end # /articles/new (GET)
  def create; end # /articles (POST)
  def show; end # /articles/1 (GET)
  def edit; end # /articles/1/edit (GET)
  def update; end # /articles/1 (PUT)
  def destroy; end # /articles/1 (DELETE)
end
```

Si ces 7 actions ne vous suffisent pas, sachez qu'il est possible d'en ajouter d'autres au moyen des mots-clés :member (pour une action agissant sur un objet) et :collection (pour une action agissant sur la liste d'objets) dans routes.rb. Dans l'exemple suivant, nous ajoutons une action "latest", accessible par HTTP GET, et une action "moderate" accessible par HTTP POST :

```
ActionController::Routing::Routes.draw do |map|
  map.resources :articles, :collection => { :latest => :get },
    :member => { :moderate => :post }
end
```

Les routes générées seront alors, respectivement, latest_articles_path et moderate_article_path. Notons toutefois que le modèle RESTful ne recommande pas ce genre d'ajouts: si vous constatez qu'en général les 7 actions prévues ne vous suffisent pas, votre décomposition en ressources est sans doute mal pensée.

respond_to

Nous avons montré qu'une ressource pouvait être représentée dans différents formats (html, xml, rss, ...). Mais comment gérer cela dans notre application ? respond_to est une méthode qui permet d'effectuer un algorithme différent (typiquement, le rendu) en fonction du format de ressource désiré :

```
def show
  @project = Project.find(params[:id])
  respond_to do |format|
    format.html
    format.xml { render :xml => @project.to_xml }
  end
end
```

Dans notre exemple, nous n'avons pas spécifié le rendu désiré pour html; Rails cherchera donc par défaut un fichier "show.html.erb". Pour le xml, nous avons spécifié que nous voulions un rendu du contenu du modèle ActiveRecord.

Ressources imbriquées

Dans notre exemple de blog, nous voulons préciser que chaque article peut être doté de commentaires. Il s'agit là d'un type de ressource supplémentaire, ayant une relation d'appartenance avec notre ressource article. Pour ce faire, on modifie config/routes.rb :

```
ActionController::Routing::Routes.draw do |map|
  map.resources :articles do |article|
    article.resources :comments
  end
end
```

Un url tel que <http://localhost:3000/articles/24/comments/1> pointera dès lors vers le 1er commentaire de l'article 24.

Pour une ressource singleton, on utilise "resource" au singulier; par exemple, pour indiquer l'auteur de chaque article (accessible via /articles/:article_id/author) :

```
ActionController::Routing::Routes.draw do |map|
  map.resources :articles do |article|
    article.resource :author
  end
end
```

Enfin, il est possible de rendre tout ceci encore plus concis avec has_one et has_many :

```
ActionController::Routing::Routes.draw do |map|
  map.resources :articles, :has_one => :author, :has_many => :comments
end
```

Echafaudage

Depuis Rails 2.0, l'échafaudage (scaffolding) propose une architecture REST par défaut. Lors d'un appel tel que :

```
script/generate scaffold article title:string body:text published:boolean
```

Rails générera automatiquement le modèle Article et sa migration, ainsi également son contrôleur RESTful et les vues correspondantes.

L'échafaudage est probablement la manière la plus facile de s'habituer au développement RESTful dans Rails. L'ancien générateur, qui produisait auparavant un échafaudage non RESTful, est maintenant disponible sous forme de plug-in.

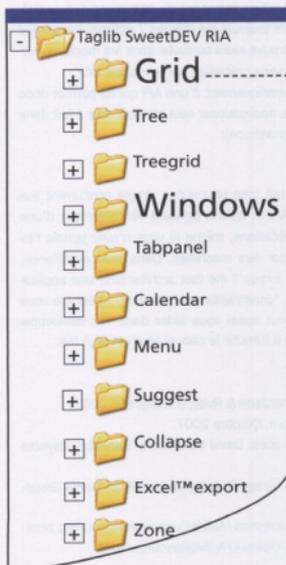
Et l'authentification ?

Pour gérer l'authentification, il y a le plug-in "restful_authentication" qui fournit toutes les méthodes nécessaires pour créer des utilisateurs et les authentifier. Le plug-in dispose d'un générateur qui produit automatiquement l'ensemble des fichiers pour les deux ressources concernées

SweetDEV RIA

Piochez dans nos tags Ajax pour développer vos interfaces Web 2.0 !

SweetDEV RIA™ est une **bibliothèque de composants Ajax**, prêts à l'emploi, pour le développement d'applications Web "riches".



Glisser/déplacer les colonnes, tri multi-colonnes, cases à cocher, pagination, scrolling horizontal, contenus multiples dans les cellules, gestion de préférences...

Gérer des composants "Window" au sein d'une structure "Layout" pour définir des IHM comparables à celles de Netvibes™ ou iGoogle™.

Pourquoi choisir SweetDEV RIA ?

- La simplicité d'utilisation
- Le support technique en France
- Le développement de composants sur mesure
- Une expertise en architecture et ergonomie
- Des formations
- Un projet Open Source

JSP

Ajax

Web 2.0

J2EE

Licence Apache 2

Téléchargez la dernière version
www.ideotechnologies.com

sweet
DEVRIA

Gros Plan

par un système d'authentification classique : les utilisateurs et les sessions. La démarche est la suivante :

```
script/plugin install http://svn.techno-weenie.net/projects/plugins/restful_authentication
script/generate authenticated user sessions
rake db:migrate
```

Ensuite, il faut rajouter les ressources dans le fichier routes.rb :

```
map.resources :users, :sessions
```

A partir de ce moment, un utilisateur peut s'inscrire via "/users/new" et se logger via "/sessions/new".

Quoi de neuf avec Rails 2.0 ?

L'essentiel des fonctionnalités RESTful a été introduit avec Rails 1.2. Toutefois, Rails 2.0 présente quelques nouveautés, qui confirment l'orientation progressive du framework vers ce type d'architecture. Nous avons vu que l'échafaudage proposé par Rails génère désormais par défaut des contrôleurs RESTful. De plus, une nouvelle tâche "rake routes" permet de lister la totalité des routes générées par votre application, ce qui est particulièrement utile dans un modèle RESTful où, comme nous l'avons vu, chaque "map.resource" génère automatiquement de nombreuses routes.

En outre, diverses améliorations dans les vues permettent de se simplifier la vie (vous en avez un aperçu dans le code généré par l'échafaudage). Ainsi, par exemple, vous pouvez désormais simplifier

```
link_to article_url(@article)
```

en

```
link_to @article
```

Enfin, Rails 2.0 facilite désormais l'authentification HTTP, qui sort du cadre de cet article, mais qui est évidemment dans l'esprit de REST.

Développement de clients RESTful

Une fois votre application REST développée, il est possible d'écrire un client capable de "consommer" son API, c'est-à-dire accéder à l'interface de manière programmée. Pour ce faire, plusieurs manières existent ; en voici deux :

Utilisation de cURL en ligne de commande

L'argument -X dans cURL permet de spécifier l'action HTTP désirée :

```
curl -X DELETE http://localhost:3000/articles/1
```

L'argument -d permet de définir des variables comme si elles étaient POSTées dans un formulaire :

```
curl -d "user[firstname]=Jean" -d "user[lastname]=Dupont" http://localhost:3000/users
```

L'argument -H permet de définir un en-tête HTTP telle que "Content-Type" ou "Accept". Cela permet par exemple de récupérer une ressource au format XML :

```
curl -H "Accept: text/xml" http://localhost:3000/articles/1"
```

Il est alors possible de parser le XML reçu avec la librairie XML de votre choix.

Utilisation de ActiveRecord

ActiveResource est un module, encore en plein développement, qui simplifie l'usage côté client de ressources REST d'une application Rails. Son comportement ressemble à s'y méprendre à celui de ActiveRecord. La seule exigence est de spécifier l'attribut "site" qui indique l'URI de l'application Rails :

```
class Project < ActiveRecord::Base
  self.site = "http://localhost:3000"
end
```

La classe ainsi définie peut-être utilisée de manière semblable à une classe ActiveRecord, et c'est ActiveResource qui s'occupe de traduire l'ensemble des instructions en requêtes HTTP :

```
projects = Project.find(:all)
new_project = Project.new(:name => "My new project")
new_project.save
```

Intérêts

Les intérêts d'une architecture REST dans une application Rails sont multiples :

- les contraintes de l'architecture imposent de conserver des contrôleurs très légers, possédant toujours les mêmes 7 actions ; l'essentiel de la logique métier se retrouve alors confinée dans les modèles eux-mêmes, ce qui va dans le sens préconisé par le modèle MVC ;
- l'application dispose automatiquement d'une API qui lui permet donc d'être utilisée par d'autres applications ; cela s'inscrit tout à fait dans la mouvance du Web 2.0 (mashups) ;

Conclusion

Nous avons vu que REST était bien plus qu'un simple concurrent aux web services tels que SOAP, et qu'il s'agissait véritablement d'une manière de penser vos applications, même si vous n'avez jamais l'intention de la faire utiliser par des machines. Dans mon expérience, REST est une véritable aide lorsqu'il me faut architecturer une application : comme le disait DHH, "constraints are liberating". J'espère vous avoir convaincu que cela peut aussi vous aider dans vos développements, et je vous encourage à franchir le cap, si ce n'est déjà fait.

Références

- RESTful Web Services, Richardson & Ruby, O'Reilly, Mai 2007.
- RESTful Rails, PeepCode.com, Octobre 2007.
- Discovering a world of resources, David Heinemeier Hansson, Keynote RailsConf 2006.
- The Future of Web Services, Gregg Pollack, Orlando Ruby Users Group, 13 décembre 2007.
- <http://api.rubyonrails.com/classes/ActionController/Resources.html>
- <http://api.rubyonrails.com/classes/ActiveResource.html>

■ Nicolas Jacobus

Co-fondateur de Belighted (www.belighted.com), une société belge de consultation et formation aux technologies Web 2.0, et Ruby on Rails en particulier. Co-blogueur sur Frailers.net, une communauté en ligne dédiée aux développeurs Ruby on Rails francophones.



Rails et la sécurité

La sécurité des applications web est souvent un sujet délicat : peu de temps à y consacrer, mais cela peut avoir des conséquences assez graves. Ruby on Rails est bien armé et avec un peu de rigueur, on peut se protéger sans trop de difficultés.

Injections SQL

Une injection SQL [1] consiste simplement à envoyer des données non prévues dans une requête SQL. Prenons comme exemple une application web où les utilisateurs sont authentifiés de la manière suivante :

```
@current_user = User.find(:first, :conditions => "login='#{@params[:login]}' AND password='#{@params[:password]}'")
```

En temps normal, quand Joe s'authentifie, la requête SQL suivante est exécutée :

```
SELECT * FROM users WHERE login='Joe' AND password='0521bc575b0ff01daa62494c70e9c5b6' LIMIT 1;
```

Mais supposons maintenant que Kevin, un Script Kiddie, passe dans le coin et décide de mettre "Joe'; -" dans le champ login. La requête SQL va alors ressembler à :

```
SELECT * FROM users WHERE login='Joe';-' AND password='000000000000000000000000' LIMIT 1;
```

Kevin a réussi à se faire passer pour Joe sans connaître son mot de passe ! Heureusement, Active Record permet de nous en protéger assez facilement [2]. Il suffit d'utiliser les formes échappées comme cela :

```
@current_user = User.find(:first, :conditions => ["login=? AND password=?", @params[:login], @params[:password]])
```

ou de façon équivalente :

```
@current_user = User.find(:first, :conditions => {:login => @params[:login], :password => @params[:password]})
```

Dans les 2 cas, Active Record rajoutera un caractère '\' devant chaque apostrophe de façon à éviter les injections SQL.

L'authentification et la gestion des droits

Pour la grande majorité des projets, l'authentification (et la gestion des droits qui vont avec) est un passage obligé. Pour cela, il existe un certain nombre de points importants à respecter comme le chiffrement des mots de passe stockés en base de données (ce que nous n'avons pas fait dans l'exemple précédent). Les erreurs sont vite arrivées, aussi je vous recommande d'utiliser des plug-in reconnus comme Restful Authentication[3], OpenID Authentication[4] et Authorization[5]. Il ne vous reste plus qu'à faire attention à un dernier détail : mettre en cache des pages nécessitant une authentification est une mauvaise idée. En effet, ces pages vont alors être servies par le serveur web sans passer par Rails, et donc sans vérification de l'authentification.

Se protéger des données forgées

L'étape suivante consiste à bien sécuriser l'accès aux données, aussi bien en lecture qu'en écriture. En effet, Rails possède quelques raccourcis très pratiques, mais qui peuvent poser problème quand ils sont mal maîtrisés. Le plus courant est l'affectation de masse [6], technique qui consiste à créer un objet Active Record directement depuis les para-

mètres de la requête HTTP. Par exemple, la création d'un compte utilisateur pourra s'effectuer de la façon suivante :

```
@user = User.create(@params[:user])
```

Supposons maintenant que la table 'users' comporte un champ 'admin' qui vaut 0 par défaut ou 1 pour les super-utilisateurs. Un utilisateur malveillant pourrait forger la requête HTTP pour ajouter un paramètre user[admin]=1 afin de gagner les pouvoirs réservés aux admins. La première solution pour se protéger de cette attaque consiste à écrire explicitement quels sont les paramètres autorisés :

```
@user = User.create(
  :login => @params[:user][:login],
  :email => @params[:user][:email],
  :password => @params[:user][:password],
  :cgu => @params[:user][:cgu])
```

Mais ceci peut vite devenir pénible quand on commence à avoir des formulaires un peu conséquents. C'est pourquoi on lui préfère généralement la deuxième solution : la déclaration dans le modèle de la liste des attributs qui ne peuvent pas être modifiés. Cette déclaration se fait à l'aide de la méthode attr_protected comme suit :

```
class User < ActiveRecord::Base
  attr_protected :admin
  ...
end
```

Nous pouvons de nouveau utiliser l'affectation de masse sans craindre qu'un utilisateur se fasse passer pour un admin, Rails s'occupe de filtrer les paramètres. Dans le même style, un attaquant peut essayer de forger des URL [7]. Si, par exemple, l'utilisateur authentifié veut supprimer l'item n°123 qui lui appartient, en appelant l'URL /items/delete/123, alors que se passera-t-il s'il appelle la même URL pour l'item n°456 qui ne lui appartient pas ? La réponse dépend du code de la méthode delete. Une implémentation de base pourrait ressembler à :

```
class ItemsController < ApplicationController
  def delete
    @item.delete(@params[:id])
  end
end
```

Pour se protéger des URL forgées, on pourrait transformer en :

```
class ItemsController < ApplicationController
  def delete
    @item = @current_user.items.find(@params[:id])
    @item.delete if @item
  end
end
```

Ce n'est pas parfait (on pourrait vérifier que c'est bien une requête de type POST), mais c'est déjà beaucoup mieux. Un dernier petit truc pour la



route avant de passer à autre chose. Si vous avez une API pour laquelle vous utilisez la sérialisation XML, il peut être intéressant de surcharger `ActiveRecord#to_xml` pour que le champ `secret_field` n'y apparaisse pas :

```
class Item < ActiveRecord::Base
  def to_xml(options={})
    super[:except => [:secret_field]].merge(options)
  end
end
```

Cross-Site Scripting

Jusqu'à maintenant, nous avons vu des attaques directes : un utilisateur essaye de s'en prendre à notre site. Il existe également des attaques plus pernicieuses que l'on classe sous le nom de Cross-Site Scripting [8] (XSS en abrégé). Leur but est de s'en prendre aux utilisateurs de notre site en glissant des "cochonneries" sur notre site. Ceci peut aller du spammeur qui mettra une balise `<iframe>` vers son site dans tous les formulaires qui lui passent sous la main à l'injection de javascript non maîtrisé. Par exemple, quelqu'un crée un item dont la description est la suivante :

```
<script>document.location='http://www.programmez.com/';</script>
```

Si maintenant un visiteur affiche la description de cet item, il sera redirigé vers le site www.programmez.com. Vous vous dites que c'est ennuyeux mais pas bien méchant ? Oui, mais la même technique permet de voler les cookies et donc les sessions associées. Nous allons donc chercher à nous protéger de ces faillies XSS.

Il est important de faire une distinction entre 2 cas est-ce que le champ que vous allez afficher peut contenir des balises HTML ou non ?

Pour afficher le nom d'un item, on sera dans le premier cas, à savoir : pas de balises HTML : on veut juste afficher le nom tel que l'a rentré son propriétaire.

Par contre, on peut souhaiter être plus souple pour la description de l'item et laisser la possibilité d'avoir un titre (balise `<h1>`), du gras (``) ou de l'italique (`<i>`).

Ces 2 cas ne se traitent pas de la même façon. Pour le premier, Rails nous offre un moyen simple de nous en protéger : le helper `h`. En pratique, à chaque fois que l'on souhaitera afficher le titre d'un item, on procédera de la manière suivante :

```
<%= h @item.title %>
```

Ce h va convertir les caractères qui pourraient être interprétés par un navigateur web en l'entité HTML correspondante. Le deuxième cas est par contre plus difficile à traiter. Vous pouvez être tenté d'utiliser un moteur de formatage de texte comme `RedCloth` [9]. Attention, cela ne suffit pas à filtrer toutes les attaques [10] ! Pour votre tranquillité, il vaut mieux utiliser le plugin `WhiteList` [11]. Depuis Rails 2.0, ce plugin fait partie du framework et peut s'utiliser de la façon suivante :

```
<%= sanitize @item.description, :tags => %w[ b i h ] %>
```

Il est possible de déclarer les balises autorisées de manière globale : je vous renvoie à la documentation officielle [12]. Et pour ceux qui veulent être sûrs de ne pas oublier d'appeler `h` ou `sanitize`, il existe des moteurs de template alternatifs comme `Safe ERB` [13] ou `Erubis` [14]. Ces moteurs adoptent l'approche opposée : ils filtrent par défaut tous les éléments `<%= %>`, charge au développeur d'indiquer explicitement ceux pour lequel le moteur ne fera pas de filtrage.

Cross-Site Request Forgeries

Avant de conclure, je voudrais juste dire un mot sur un dernier type d'attaques. Les CSRF, abréviation de Cross-Site Request Forgery, sont des attaques complexes qui visent à forcer l'utilisateur à envoyer une requête HTTP vers notre site lorsque celui-ci visitera le site de l'attaquant. Je vous renvoie à wikipedia [15] si vous voulez comprendre comment fonctionne ce type d'attaques. Sachez que Rails vous protège de celles-ci depuis la version 2.0 et qu'il existe un plugin pour les versions plus anciennes : `CSRF-killer` [16].

Conclusion

Nous avons pu voir qu'en prenant quelques bonnes habitudes, on pouvait développer des applications sûres en Rails. Il reste d'autres sujets comme l'administration de Rails [17], Apache [18] ou MySQL [19][20], la manipulation des fichiers [21] ou encore le filtrage des informations sensibles dans les logs [22]. Et tenez-vous au courant des mises à jour de sécurité de Rails et des plugin utilisés. Je remercie **Benoit Sibaud** pour sa relecture attentive.

Références

- [1] http://fr.wikipedia.org/wiki/Injection_SQL
- [2] <http://manuals.rubyonrails.com/read/chapter/43>
- [3] http://agilewebdevelopment.com/plugins/restful_authentication
- [4] <http://agilewebdevelopment.com/plugins/openidauthentication>
- [5] <http://www.writertopia.com/developers/authorization>
- [6] <http://manuals.rubyonrails.com/read/chapter/47> (Creating records directly from form parameters)
- [7] <http://www.the railsway.com/2007/3/26/association-proxies-are-your-friend>
- [8] http://fr.wikipedia.org/wiki/Cross_site_scripting
- [9] <http://whytheluckystiff.net/ruby/redcloth/>
- [10] <http://www.rorsecurity.info/2007/08/20/redcloth-security-thoughts/>
- [11] http://svn.techno-weenie.net/projects/plugins/white_list/
- [12] <http://api.rubyonrails.org/classes/ActionView/Helpers/SanitizeHelper.html#M000936>
- [13] <http://wiki.rubyonrails.com/rails/pages/Safe+ERB>
- [14] <http://www.kuwata-lab.com/erubis/>
- [15] http://fr.wikipedia.org/wiki/Cross-Site_Request_Forgeries
- [16] http://svn.techno-weenie.net/projects/plugins/csrf_killer/
- [17] <http://www.igvita.com/blog/2006/10/10/securing-your-rails-environment/>
- [18] <http://www.rorsecurity.info/2007/03/15/apache-2-file-privileges-and-modules/>
- [19] <http://www.rorsecurity.info/2007/02/25/securing-mysql/>
- [20] <http://www.rorsecurity.info/2007/02/27/rails%e2%80%99-friends-securing-mysql-continued/>
- [21] <http://www.rorsecurity.info/2007/03/27/working-with-files-in-rails/>
- [22] <http://railscasts.com/episodes/9>
- [23] <http://www.rorsecurity.info/ruby-on-rails-security-cheatsheet/>
- [24] <http://www.quarkruby.com/2007/9/20/ruby-on-rails-security-guide>
- [25] <http://blog.innerewut.de/2008/1/3/24c3-ruby-on-rails-security>
- [26] http://www.owasp.org/index.php/OWASP_AppSec_FAQ
- [27] http://www.owasp.org/index.php/Top_10_2007

■ Bruno Michel

Ingenieur diplômé de l'IECNAM. Développeur chez AF83 - <http://www.af83.com/>. Il s'occupe du site <http://linuxfr.org> et participe à deux associations l'April et Ruby France. bruno.michel@af83.com.

École Centrale d'Électronique : miser sur l'embarqué !

L'ECE, même si elle n'est pas la plus connue des écoles, possède des atouts indéniables pour les étudiants en informatique et notamment ceux intéressés par l'informatique embarquée et la programmation. Faisons un peu mieux connaissance avec elle.

Comme d'autres établissements de même catégorie, l'ECE dispense une formation d'ingénieur en 5 ans après le bac S sur le modèle Licence - Master. Elle est reconnue et habilitée à délivrer le titre d'ingénieur. Les disciplines enseignées à l'école sont l'informatique, l'électronique, les systèmes embarqués, les télécoms et réseaux, la finance quantitative, le multimedia et le web. En 2^e année du cycle ingénieur, l'étudiant choisit librement une spécialisation (majeure) parmi les trois proposées : systèmes d'information, systèmes embarqués et télécommunications-réseaux. " L'objectif est de former des ingénieurs logiciels connaissant différents langages, avec une capacité d'analyse et une



forte maîtrise technologique " explique Justin Templemore, responsable des enseignements de systèmes d'information. Durant les trois premières années, on enseigne le langage C et les notions objets appliquées à travers de Java. Ensuite " l'objet " devient primordial (avec du codage en C#, Java notamment). Si le cycle se déroule en France, l'établissement met un point d'honneur à ouvrir l'étudiant vers l'international avec des séjours obligatoires sur des campus nord-américains, dans des établissements partenaires et des semestres d'études en dernière année (accords signés avec une trentaine d'universités étrangères). Cela permet d'être polyvalent.

L'embarqué comme priorité

L'univers de l'enfoi, de l'embarqué est très important pour l'ECE. Les étudiants peuvent

ainsi découvrir et devenir des experts sous Windows CE, .Net Compact Framework, et en Java embarqué. L'enseignement est de très haut niveau et cela passe aussi par une maîtrise des notions UML.

Au-delà de cet univers, l'architecture n'est pas oubliée, bien au contraire, l'ECE souhaite sensibiliser l'étudiant à ces problématiques. Un autre point important : la veille technologique. Elle n'est pas liée directement aux cours mais plutôt aux associations étudiantes gravitant dans l'école. Elles permettent de faire le lien entre les cours et l'évolution technologique et surtout, de tenir au courant l'étudiant. Les associations (comme la Iteam) sont idéales pour concevoir des projets informatiques et en proposer soi-même. Pour ne pas isoler le cursus de l'entreprise, des experts de l'industrie interviennent en 4^e et en 5^e année. Des éditeurs comme Microsoft et Oracle sont présents et d'autres collaborations sont en cours.

L'admission

À l'instar d'autres écoles d'ingénieurs, l'admission se fait sur concours après un Bac S, une CPGE, un DUT ou une Licence. En 3^e année, les étudiants partent à l'étranger, notamment à Montréal où les cours sont dispensés en anglais. " On a aussi des nocturnes et des postes en libre accès pour nos étudiants, " précise Justin Templemore.

L'école est plutôt Windows mais Linux se fait aujourd'hui une place non négligeable sur les postes et les cours. Cependant, l'étudiant est libre de choisir. Et selon les cours et les projets montés par les professeurs, l'établissement fournit matériels et logiciels nécessaires. La journée type se déroule sur 6 heures de cours, avec 20 à 25 heures de présence par semaine. Du temps libre est prévu dans l'agenda pour réaliser des projets professionnels, personnels, ou participer aux associations. 50 % du temps est consacré à la réalisation de projets. Il faut être le plus concret possible ! " Nous avons également



des parrainages avec des entreprises, comme avec Sopra cette année. Des tuteurs accompagnent les étudiants pour les projets et les professeurs sont aussi très disponibles, c'est important", explique J.Templemore. Les activités extra scolaires sont encouragées par la direction, qu'il s'agisse de sport, d'humanitaire, de programmation, d'open source... " Nous avons une trentaine d'associations dont une junior entreprise. "

Et la parité ? Actuellement, le cursus accueilli 15 % de femmes. Ce n'est pas beaucoup (aucun quota n'existe). " Elles ont autant leur place ici à l'ECE que les hommes ! " s'exclame Justin Templemore. Sans doute sont-elles moins attirées par le développement.

Site web : <http://www.ece.fr>

■ François Tonlic

L'ECE en chiffres

Aujourd'hui, l'ECE, qui est implantée à Paris, est forte de 1300 étudiants, dont 860 en cycle ingénieur.

L'équipe enseignante se compose de 200 enseignants (permanents et vacataires de l'enseignement supérieur ou du monde professionnel).

- Le taux de réussite est d'environ 92 %.
- Le coût de la scolarité revient à 7000 euros.
- Quatre stages sont obligatoires (durée moyenne 12 mois)... Les étudiants se répartissent entre les trois spécialisations " majeures " à part égale et la direction n'impose pas à l'étudiant un domaine plus qu'un autre, l'actualité et l'évolution du marché peuvent influencer leur choix.

Adobe FLEX 3 et AIR



Une nouvelle ère pour le développeur

Le 25 février dernier, Adobe a rendu disponible la version 3.0 de Flex, sa plate-forme RIA (Rich Internet Application), ainsi que la version 1.0 du tout nouveau AIR pour les applications RDA (Rich Desktop Application). Il s'agit d'un virage important pour le développement Web et Desktop, avec une véritable convergence entre les deux.



Fx

Flex 3.0 renouvelle profondément la technologie Flex : langage plus puissant (ActionScript 3, couplé à MXML), ouverture d'une partie du code et des fonctions, support des dernières technologies, nouvel environnement de développement, nouveaux services serveurs. AIR est pour Adobe un nouveau défi. Pour la première fois, l'éditeur aborde pleinement les applications de bureau, tout en s'appuyant sur des technologies bien connues du développeur : HTML, ActionScript, XML, PDF, Flash. Le tout reposant sur des API et sur un environnement d'exécution disponible sur Windows, Linux, MacOS X... Bref, le cœur de l'application reste Internet mais avec des fonctions natives. La capacité multi-plate-forme est une des forces de AIR. La convergence n'est pas un vain mot pour l'éditeur car une application Flex 3 s'exécute aussi sur AIR...

Pour le développeur et le designer, Adobe met là aussi la convergence en avant. Car une application Flex et AIR exploite les fonctions de LiveCycle, de Coldfusion, de



Flash Media Server, de Connect, de Flash Cast ou encore de Scene7, tout en reposant sur des protocoles standard du Web et des Web Services.

Mais surtout, on n'est pas dépaycé par l'interface et le développement : Creative Suite, Flex Builder (s'appuyant sur Eclipse). Et le cycle de développement utilise

largement tous les outils de la gamme Creative Suite dont la pierre angulaire est le futur outil Thermo, outil de design web par excellence !

Découvrons dès à présent toutes les nouveautés de Flex 3 et de AIR, avec des exemples de code très concrets.

■ F.T.

Fx

Adobe Flex 3 : une nouvelle gam

Avant de parler des nouveautés de la version 3, voici quelques rappels : qu'est ce que la solution Flex, à qui s'adresse-t-elle et quelle est sa composition ? Flex, c'est un environnement pour développer des applications Flash multi-plates-formes qui s'exécutent dans le navigateur, sur le bureau (grâce à Adobe AIR) et demain sur les mobiles. Cet environnement a été conçu pour les développeurs. Les profils les plus à l'aise pour se former à Flex sont les codeurs Java et .Net. En effet, le langage de script ActionScript 3, basé sur le standard ECMAScript 262 requiert de bonnes notions de développement objet (héritage de classes, design pattern, MVC, gestion événementielle...). Voici la composition de l'offre Flex 3 :

- **Flash Player 9** : Côté client, ce plugin déjà déployé sur plus de 97% des ordinateurs dans le monde (Windows, Mac, Linux), embarque la nouvelle machine virtuelle AVM2 (avec un compilateur JIT). Le développeur Flex n'a donc pas à gérer les problèmes de types de navigateurs, d'OS ou de versions, ce qui diminue les phases de tests et de recettes. Cette nouvelle machine virtuelle est un projet Open Source, le projet Tamarin de la fondation Mozilla.

- **Flex 3 SDK** : Ce kit de développement est gratuit et pour la première fois, il devient aussi Open Source. Il est composé d'un langage de description d'interface en XML (le MXML) et d'un langage de script (ActionScript 3). Il est aussi livré avec un compilateur et un débogueur. Une fois compilé, un projet Flex est un fichier Flash (SWF). Il suffit de le déployer dans une page HTML sur le serveur web.

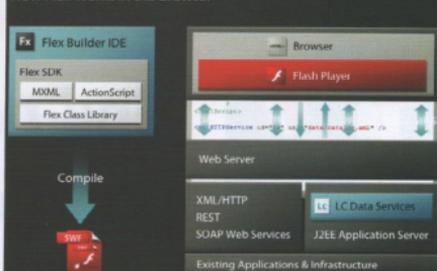
- **Flex Builder 3** : C'est l'environnement de développement proposé par Adobe. Basé sur Eclipse, il propose des fonctions avancées de debug, profiling et de conception graphique des interfaces. Gratuit pour les étudiants (www.flexregistration.com), cet IDE est disponible en deux versions : la Standard et la Professionnelle.

- **Blaze DS** : Ce composant optionnel s'adresse aux applications Flex qui communiquent avec des backends JEE. BlazeDS est un sous-ensemble gratuit et Open Source (licence LGPL) de LiveCycle Data Services. Il permet d'améliorer sa productivité en invoquant directement des services Java, et de fluidifier le trafic sur le réseau grâce au standard Open Source AMF. BlazeDS introduit aussi un moyen de réalisation du push de données par envoi de messages sur le port HTTP.

Le Deep-Linking : une nouveauté pour améliorer le référencement de vos applications Flex

Outre l'ouverture du SDK en licence MPL, voici une nouveauté de Flex 3 qui résout une problématique standard du développeur de RIA : la navigation intra-écrans, intra-composants et le référencement. Pour utiliser les fonctionnalités de Deep-Linking, il faut veiller à intégrer 3 fichiers inclus dans le SDK (history.css, history.js et historyFrame.html) et à activer le deep-linking au niveau du compilateur. Lorsqu'un internaute utilise une RIA, il est tenté de cliquer naturellement sur les boutons " précédent " et " suivant " du navigateur. Or dans une RIA, on ne raisonne pas en pages mais en écrans, onglets, vues, etc. De nombreux composants Flex savent nativement gérer cette navigation en envoyant des informations au HistoryManager du navigateur. Voici un exemple qui active cette propriété sur le composant de navigation par onglets :

How Flex Works in the Browser



```
<mx:TabNavigator historyManagementEnabled="true">
```

Grâce à ces échanges avec le navigateur, le développeur de RIA peut aussi modifier les propriétés du navigateur pour faire correspondre une vue à une URL précise. Dans cet exemple, je m'amuse à modifier l'URL du navigateur à chaque changement d'onglet et d'état d'une case à cocher.

```
<?xml version="1.0" encoding="utf-8"?>
<- deplinking/ComplexMultiPanelExample.mxml ->
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
  layout="vertical"
  historyManagementEnabled="false"
  creationComplete="init();parseURL(event)"
  >
  <mx:Script>
  <![CDATA[
    import mx.events.BrowserChangeEvent;
    import mx.managers.IBrowserManager;
    import mx.managers.BrowserManager;
    import mx.utils.URLUtil;

    private var bm:IBrowserManager;

    [Bindable]
    private var answerFromURL:Boolean;
    [Bindable]
    private var cityFromURL:String;

    private function init():void {
      bm = BrowserManager.getInstance();
      bm.addEventListener(BrowserChangeEvent.BROWSER_URL_CHANGE, parseURL);
      bm.init("", "Bienvenue les amis!");
    }
  ]]>
```

me Open Source pour développer des RIA

```
private function parseURL(event:Event):void {
    var o:Object = URLUtil.stringToObject(bm.fragment, "&");
    if (o.panel == undefined)
        o.panel = 0;

    tn.selectedIndex = o.panel;
    tn.validateNow();

    cityFromURL = o.city;
    answerFromURL = o.answerBox;

    if (o.answerBox != undefined){
        myBox.selected = o.answerBox;
    }
}

public function updateTitle():void {
    bm.setTitle("Bienvenue à " + myCity.text + "!");
}

private function updateURL():void {
    try {
        cityFromURL = myCity.text;
        answerFromURL = myBox.selected;
    } catch (e:Error) {
    }

    var o:Object = {};

    try {
        o.panel = tn.selectedIndex;
        o.city = myCity.text;
        o.answerBox = myBox.selected;
    } catch (e:Error) {
    } finally {
        var s:String = URLUtil.objectToString(o, "&");
        bm.selfragment(s);
    }
}
]]>
</mx:Script>

<mx:TabNavigator id="tn" width="300" change="updateURL()" height="148">
    <mx:Panel label="Rubrique 1" layout="absolute">
    <mx:Label text="Votre ville : " x="10" y="10"/>
    <mx:TextInput x="87" y="8" id="myCity"/>
    <mx:Button x="138" y="43" label="Change le titre" click="updateTitle"/>
    </mx:Panel>
```

```
<mx:Panel label="Rubrique 2" layout="absolute">
    <mx:Label text="Toujours d'accord ?" x="10" y="10"/>
    <mx:CheckBox x="130" y="8" label="Oui" id="myBox" click="updateURL"/>
    </mx:Panel>
</mx:TabNavigator>
</mx:Application>
```

La classe AS3 **BrowserManager** est centrale. La fonction setTitle met directement à jour le titre de la page (essentiel pour un bon référencement de vos applications Flex) et les fonctions updateURL et parseURL présentent ici des mécanismes de base d'échanges entre l'application Flex et l'URL dans le navigateur. Lors de l'initialisation de l'application, je me place à l'écoute de l'événement BROWSER_URL_CHANGE. Voici la forme que prend l'URL du navigateur :

```
appliRIA.html#panel=1&city=&answerBox=true
```

Si vous modifiez l'URL en remplaçant le dernier booléen true par false, et que vous rechargez la page, vous verrez la case se décocher comme par magie. Désormais, si vous copiez/collez cette URL dans un autre onglet du navigateur, l'application Flex se chargera en ouvrant directement le second onglet. Ces nouvelles techniques vont vous permettre de référencer des vues comme on référence plus classiquement des pages HTML.

Des nouveaux composants applicatifs

Les nouveaux composants de la suite Flex 3 proposent des expériences applicatives toujours plus riches et professionnelles. Tous les développeurs Flex sont familiers avec le composant DataGrid, tableau d'affichage et de tri de données. C'est peut-être le composant Flex le plus utilisé dans les projets Entreprise. La nouvelle classe AS3 AdvancedDatagrid offre beaucoup plus de fonctionnalités que la version classique : tri multi-colonnes, regroupement de données et affichage hiérarchique (arbre), regroupement de colonnes...

Le regroupement s'effectue directement au niveau de la source de données (le dataProvider). Les nouvelles classes de GroupingCollection permettent de simplement définir les critères de regroupement. Voici par exemple une collection qui décrit l'activité commerciale de mon équipe sur des zones géographiques :

```
[Bindable]
private var dpFlat:ArrayCollection = new ArrayCollection([
    {Region:"Europe", Territory:"France", Territory_Rep:"Barbara Jennings", Actual:38865, Estimate:40000},
    {Region:"Europe", Territory:"France", Territory_Rep:"Dana Binn", Actual:29885, Estimate:30000},
    {Region:"Europe", Territory:"Espagne", Territory_Rep:"Joe Smith", Actual:29134, Estimate:30000},
    {Region:"Europe", Territory:"Espagne", Territory_Rep:"Bethany Pittman", Actual:52888, Estimate:45000},
    {Region:"Europe", Territory:"Portugal", Territory_Rep:"Lauren Ipsum",
```

```
Actual:38805, Estimate:40000],
[Region:"Asie", Territory:"Corée", Territory_Rep:"T.R. Smith", Actual:55
498, Estimate:40000],
[Region:"Asie", Territory:"Chine", Territory_Rep:"Alice Treu", Actual:
44985, Estimate:45000],
[Region:"Asie", Territory:"Chine", Territory_Rep:"Jane Grove", Actual:
44913, Estimate:45000]
]);
```

Dans cette collection, deux regroupements sont possibles sur les colonnes Region et Territory. Si j'utilise le composant DataGrid, j'obtiens un tableau classique avec une ligne par enregistrement. Avec AdvancedDataGrid, je peux en profiter pour effectuer un travail de regroupement sur la source de données :

```
<mx:AdvancedDataGrid id="myADG"
width="100%" height="100%"
initialize="gc.refresh();">
<mx:dataProvider>
<mx:GroupingCollection id="gc" source="{dpFlat}">
<mx:grouping>
<mx:Grouping>
<mx:GroupingField name="Region"/>
<mx:GroupingField name="Territory"/>
</mx:Grouping>
</mx:grouping>
</mx:GroupingCollection>
</mx:dataProvider>
<mx:columns>
<mx:AdvancedDataGridColumn dataField="Region"/>
<mx:AdvancedDataGridColumn dataField="Territory"/>
<mx:AdvancedDataGridColumn dataField="Territory_Rep"
headerText="Territory Rep"/>
<mx:AdvancedDataGridColumn dataField="Actual"/>
<mx:AdvancedDataGridColumn dataField="Estimate"/>
</mx:columns>
</mx:AdvancedDataGrid>
```

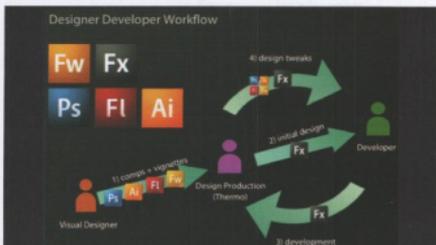
J'indique que ma source de données est la collection dpFlat et qu'il faut regrouper les champs Region et Territory. Je spécifie ensuite l'ordre des colonnes que je souhaite voir apparaître en entête. Une fois le code compilé, j'obtiens un fichier flash .swf qui affiche de façon hiérarchique la donnée : Cette classe AS3 a été étendue dans le SDK pour proposer des datagrid OLAP (online analytical processing). C'est un concept proche des tables

Region	Territory	Territory_Rep	Actual	Estimate
China	Asie	Jane Grove	44913	45000
	Asie	Alice Treu	44985	45000
	Asie	T.R. Smith	55498	40000
Europe	Europe	Jane Smith	25134	26000
	Europe	Bathory Pittman	53888	43000
France	France			
Italy	Italy			

pivots de Microsoft Excel en fournissant une vue à 2 dimensions des lignes et des colonnes pour agréger la donnée au sein d'une vue. Les représentations OLAP sont pertinentes quand vous récupérez des milliers de lignes de données et que vous souhaitez présenter une vue synthétique et faciliter l'analyse ou la prise de décision. Le processus de création d'une OLAPDataGrid est découpé en plusieurs étapes : réception de la donnée à plat, définition d'un schéma OLAP pour le cube OLAP, définition d'une requête OLAP et affichage dans le composant OLAPDataGrid.

Product	Quarter							
	2006		2007					
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
ColdFusion	770	125	375	215	715	175	410	215
Flex	210	760	210	430	210	670	100	300
DreamWeaver	470	175	840	330	310	175	1150	270
Flash	280	570	670	1010	370	620	420	1240

Des outils pour industrialiser ses développements Flex



Flex Builder 3 étoffe sa gamme d'outils et d'assistants pour améliorer la productivité du développeur Flex. J'ai particulièrement apprécié les assistants de connexion aux données que je perçois plus comme des tutoriaux de haut niveau que comme de réels générateurs de code. Le principe est simple : vous indiquez à Flex Builder quelle technologie serveur vous utilisez pour votre projet (PHP, ASP ou Java EE), puis vous spécifiez une connexion à une table de votre base de données. Il est ensuite possible de spécifier les champs de la table qui nous intéressent et Flex Builder va automatiquement générer une application CRUD en Flex pour gérer votre base : listing de la table dans une DataGrid, création de nouveaux enregistrements, mise à jour, suppression, etc. Le code est de qualité et permet de comprendre facilement les mécanismes d'appels distants (remoting) et de communication en AMF (format d'échange binaire Open Source).

L'introspection de WSDL est une nouveauté qui génère automatiquement des classes AS3 qui assurent le mapping des objets et l'invocation des méthodes du webservice. A la manière d'un Visual Studio, Flex Builder analyse le wsdl et génère automatiquement les packages de classes pour consommer des objets complexes, gérer les appels, les résultats et les erreurs des méthodes publiques. Enfin, le profiler est un autre exemple d'outil d'aide à l'amélioration de la qualité de son code. Cette nouvelle perspective de l'IDE permet d'analyser en temps réel la charge mémoire, de détecter d'éventuelles fuites ou des pics d'utilisation de la mémoire côté client. Associé au débogger pas à pas, le profiler deviendra un assistant indispensable du flexeur professionnel.

Ma première application AIR avec Flex Builder

Le développement d'applications riches bureautiques est désormais intégré dans l'environnement Flex Builder. Le principe est de coder une application avec le Flex SDK et de la compléter avec les classes AS3 du AIR SDK pour l'exécuter dans AIR. L'expérience de programmation dans Flex Builder est très simple, voici comment créer sa première application AIR.

Une fois Flex Builder lancé, il suffit de créer un nouveau projet Flex (File > New > Flex Project). L'IDE vous demande alors si vous souhaitez une application qui s'exécute dans le navigateur (RIA Flex) ou sur le bureau (RDA grâce à AIR). Sélectionnez la seconde option, donnez un nom au projet, "monAppli" par exemple, et cliquez sur le bouton "Finish". Flex Builder crée automatiquement la structure du projet AIR et deux fichiers attirent l'attention : monAppli.mxml (le main file de mon projet) et monAppli-app.xml qui est le manifeste qui décrit l'identification, l'installation et le lancement de l'application AIR. Le fichier monAppli.mxml débute avec un nouveau tag, WindowedApplication, que vous pouvez enrichir d'un titre.

```
<?xml version="1.0" encoding="utf-8"?>
<mx:WindowedApplication xmlns:mx="http://www.adobe.com/2006/
/mxml" layout="absolute" title="mon Application">
</mx:WindowedApplication>
```

La vue Design de Flex Builder ne permet de rapidement créer des écrans et d'agencer les composants de mon interface riche. De nouveaux composants spécifiques à AIR sont disponibles. Vous pouvez par exemple rajouter un label classique, mais aussi un nouveau composant qui interprète le HTML (grâce au composant webkit) et un composant fileTree pour afficher l'arborescence de vos fichiers en local.

Voici le code qui en résulte :

```
<?xml version="1.0" encoding="utf-8"?>
<mx:WindowedApplication xmlns:mx="http://www.adobe.com/2006/
/mxml" layout="absolute" title="mon Application">
<mx:Label x="10" y="10" text="Bienvenue dans mon application."/>
<mx:HTML x="180" y="36" width="372" height="416" location="http://
www.adobe.fr"/>
<mx:FileSystemTree x="10" y="36" height="416"/>
</mx:WindowedApplication>
```



Vous obtenez par défaut une fenêtre aux couleurs de votre OS que vous pouvez réduire, maximiser et fermer. Pour créer une fenêtre principale qui ne reprend pas le look & feel de l'OS, rajoutez un tag de Style pour rendre légèrement transparent le fond de l'application.

```
<mx:Style>
WindowedApplication
{
background-color:"0x999999";
```

```
background-alpha:"0.5";
}
</mx:Style>
```

Si vous compilez l'application, vous constaterez que cela n'a pour l'instant aucun impact. Il faut donc éditer le manifeste monAppli-app.xml et indiquer que vous souhaitez un look de fenêtre (chrome) indépendant de l'OS et gérer la transparence. Recherchez, décommentez et éditez ces attributs dans le manifeste XML :

```
<!-- The type of system chrome to use (either "standard" or "none").
Optional. Default standard. -->
<systemChrome>none</systemChrome>

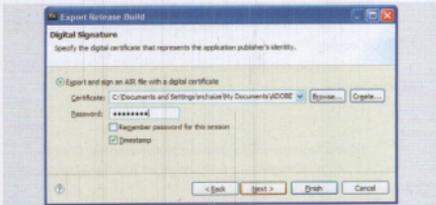
<!-- Whether the window is transparent. Only applicable when system
Chrome is false. Optional. Default false. -->
<transparent>true</transparent>
```



Lancez l'application (CTRL + F11). Le fond de la fenêtre est bien transparent.

Tout fonctionne, reste à créer un package d'installation .air pour distribuer l'application.

Dans le menu Project, lancez la fonction "Export Release Build". Spécifiez l'application AIR à packager et cliquez sur suivant. Il faut impérativement signer une application AIR. Si vous ne possédez pas de certificat numérique, vous pouvez en créer un en quelques secondes grâce à Flex Builder.



Cliquez sur Finish pour créer le package signé. Par défaut, Flex Builder va placer le package monAppli.air dans le répertoire du projet Flex. Double-cliquez sur le package, une phase d'installation classique d'une application AIR démarre. Je vous invite à parcourir le manifeste monAppli-app.xml pour découvrir comment personnaliser l'icône, le nom de l'application, les informations sur l'auteur de l'application, le copyright, etc.



ADOBE® AIR : une nouvelle

AIR (Adobe Integrated Runtime) est un environnement d'exécution multi-plate-forme qui permet d'exécuter vos applications web (développements AJAX, Flex ou Flash) sur le bureau des utilisateurs, en dehors du navigateur. Ce n'est pas un nouveau langage, c'est un nouveau lieu d'expression pour vos applications riches. Pour un développeur Flex, le AIR SDK ne va rajouter que quelques classes AS3 propres au monde des applications riches bureautiques : accès aux fichiers locaux, gestion du multi-fenêtrage, gestion de la base locale SQLite, glisser-déposer avec le bureau, gestion de la déconnexion, etc. Adobe AIR est composé de plusieurs projets Open Source : le Flash Player 9 et sa machine virtuelle (projet Tamarin), une base de données SQLite et un composant d'affichage de HTML (projet WebKit). Une application riche qui s'exécute dans AIR peut allier facilement Flash, Html et PDF. Si vous ne l'avez pas encore installée, je vous invite à tester l'Ebay Desktop (<http://desktop.ebay.com>). C'est une application AIR qui révolutionne la navigation et la gestion de son activité sur eBay. Du point de vue de la programmation, voici quelques exemples d'applications AIR réalisées directement dans le Flex Builder 3.

Glisser-Déposer : du bureau vers AIR

Un des points qui me frustrent souvent avec les RIA dans un navigateur c'est l'impossibilité de glisser-déposer des fichiers directement depuis mon poste, vers l'application (sans installation de plug-in ActiveX bien sûr). AIR offre plus de libertés et la gestion du drag and drop est purement événementielle. Voici un exemple d'application AIR où l'utilisateur peut glisser un ou plusieurs fichiers sur la partie haute de l'application. Les noms des fichiers s'affichent dans la partie basse.

```
<?xml version="1.0" encoding="utf-8"?>
<mx:WindowedApplication xmlns:mx="http://www.adobe.com/2006/
mxm1" layout="absolute" creationComplete="initApp()">
  <mx:Script>
    <![CDATA[
      import mx.controls.Alert;
      private function initApp():void{
        monLab.addEventListener(NativeDragEvent.NATIVE_DRAG
        _ENTER,onDragIn);
        monLab.addEventListener(NativeDragEvent.NATIVE_DRAG
        _DROP,onDropIn);
      }

      private function onDragIn(event:NativeDragEvent):void{
        NativeDragManager.dropAction = NativeDragActions.
COPY;
        if(event.clipboard.hasFormat(ClipboardFormats.FILE_LIST
FORMAT)){
          NativeDragManager.acceptDragDrop(monLab);
        }else{
          mx.controls.Alert.show("Interdit. Vous ne pouvez déposer
que des fichiers.");
        }
      }
    ]]>
  </mx:Script>
</mx:WindowedApplication>
```

```

    }
  }
  private function onDropIn(event:NativeDragEvent):void{
    var arr:Array = event.clipboard.getData(ClipboardFormats.
FILE_LIST_FORMAT) as Array;
    var f:File = File(arr[0]);
    for each (var item:File in arr)
    {
      maZoneTexte.text += item.name + "\n";
    }
  }
]}>
</mx:Script>
<mx:Panel x="10" y="10" width="250" height="200" layout="absolute"
title="Drag IN">
  <mx:TextArea x="10" y="82" width="210" height="68" id
="maZoneTexte"/>
  <mx:TextArea x="10" y="10" width="210" height="64" text=
"Glissez vos fichiers ici !" textAlign="center" backgroundColor="#B
C E0FF" fontWeight="bold" id="monLab" editable="false"/>
</mx:Panel>
</mx:WindowedApplication>
```

Lors de l'initialisation de l'application, ma zone de texte monLab est à l'écoute de deux événements : *Native_drag_enter* et *Native_drag_drop*. Lorsqu'un fichier est glissé-déposé depuis le bureau de l'utilisateur, et qu'il passe sur le composant *monLab*, la fonction *onDragIn* est appelée et analyse le contenu du presse-papier (objet *Clipboard*). Cela permet de tester le type de données qui transite en mémoire : texte, binaire, url, tableau de fichiers... Je n'autorise le Drag and Drop sur le composant *monLab* que si l'on y glisse des fichiers (fonction *acceptDragDrop*). Au moment où l'utilisateur lâche son lot de fichiers sur le composant, je parcours le tableau de résultat pour récupérer le nom de chaque fichier et je les affiche dans le composant texte *maZoneTexte*.

La gestion d'une déconnexion avec Adobe AIR

Beaucoup d'utilisateurs souhaitent pouvoir lancer à tout moment leurs applications web et retrouver la même expérience utilisateur sans connexion internet.

Adobe AIR fournit les événements au développeur pour repérer si un utilisateur est connecté ou non au service internet. En cas de déconnexion, l'utilisateur peut tout de même lancer l'application, profiter de la base de données locale SQLite et synchroniser son travail une fois la connexion retrouvée. Voici un exemple d'application AIR qui affiche si un utilisateur est connecté ou non au service internet, et si ce statut on-line/offline change.

ère pour vos applications web

```
<?xml version="1.0" encoding="utf-8"?>
<mx:WindowedApplication xmlns:mx="http://www.adobe.com/2006/
mxml" layout="absolute" creationComplete="init">
  <mx:Script>
    <![CDATA[
      import air.net.URLMonitor;
      import air.net.SocketMonitor;
      import flash.events.StatusEvent;

      [Bindable]
      public var statusText:String;

      public var netMon:URLMonitor;

      public function StatusChange(e:StatusEvent):void {
        statusText = "Current status: " + netMon.available;
        switch [netMon.available]{
          case true:
            symbol_disconnected.visible = false;
            symbol_connected.visible = true;
            statusText = "Network available";
            break;
          default:
            symbol_disconnected.visible = true;
            symbol_connected.visible = false;
            statusText = "Network unavailable";
            break;
        }
      }

      public function init():void{
        symbol_disconnected.visible = true;

        netMon = new URLMonitor(new URLRequest("http://www.
adobe.com/"));
        netMon.addEventListener(StatusEvent.STATUS, Status
Change);
        netMon.start();
      }
    ]]>
  </mx:Script>
  <mx:Text text="{statusText}" horizontalCenter="0" fontWeight="bold"
fontFamily="Courier New" fontSize="16" bottom="0"/>
  <mx:Image id="symbol_disconnected" horizontalCenter="0" vertical
Center="0" width="125" height="125" source="assets/imgs/disconnected
.png" visible="false"/>
  <mx:Image id="symbol_connected" horizontalCenter="0" verticalCenter
="0" width="125" height="125" source="assets/imgs/connected.png"
visible="false"/>
</mx:WindowedApplication>
```

En fonction de l'état de connexion de l'utilisateur, on va faire apparaître ou masquer une image PNG. L'objet URLMonitor surveille une URL sur le réseau (ici le site Adobe mais vous pouvez imaginer de surveiller votre intranet ou un service distant). On place cet objet à l'écoute d'un événement STATUS qui est envoyé à chaque changement d'état de la connexion. Lors d'un changement d'état de connexion, la fonction StatusChange analyse si l'utilisateur est online ou offline. Une fois de plus le modèle évènementiel de Flex suffit à gérer cette problématique d'applications bureautiques en quelques lignes de code.

Une base de données locale SQLite

SQLite est un projet de base de données Open Source. Adobe vient de rentrer officiellement dans le consortium pour contribuer au projet plus efficacement. Les bases SQLite ont l'avantage d'être très rapides d'accès et réunissent toutes les informations dans un seul fichier par base. Voici une application AIR qui permet de consulter la base locale, de rajouter des enregistrements et d'effectuer des recherches :

```
<?xml version="1.0" encoding="utf-8"?>
<mx:WindowedApplication xmlns:mx="http://www.adobe.com/2006/
mxml" title="Contact Manager" creationComplete="init" layout="
absolute">
  <mx:Script>
    <![CDATA[
      private var dbConn:SQLConnection;

      private function init():void
      {
        tab_nav.addEventListener(Event.CHANGE, function():void {
          if(tab_nav.selectedIndex == 0) listContacts();
        });

        var dbFile:File = File.applicationDirectory.resolvePath
("contacts.db");
        dbConn = new SQLConnection();
        var dbStatement:SQLStatement = new SQLStatement();
        dbStatement.sqlConnection = dbConn;
        dbConn.open(dbFile);
        dbStatement.text = "CREATE TABLE IF NOT EXISTS
contacts (name TEXT, lastname TEXT, email TEXT)";
        dbStatement.execute();
        listContacts();
      }

      private function listContacts():void
      {
        var dbStatement:SQLStatement = new SQLStatement();
        dbStatement.sqlConnection = dbConn;
        dbStatement.text = "SELECT * FROM contacts";
        dbStatement.addEventListener(SQLEvent.RESULT, listContacts
```

```

Result]);
        dbStatement.execute();
    }

    private function listContactsResult(evt:SQLEvent):void
    {
        list_dg.dataProvider = evt.target.getResult().data;
    }

    private function addContact():void
    {
        var dbStatement:SQLStatement = new SQLStatement();
        dbStatement.sqlConnection = dbConn;
        dbStatement.text = "INSERT INTO contacts VALUES (:NAME,
:LASTNAME, :EMAIL)";
        dbStatement.parameters[":NAME"] = name_txt.text;
        dbStatement.parameters[":LASTNAME"] = lastname_txt.text;
        dbStatement.parameters[":EMAIL"] = email_txt.text;
        dbStatement.execute();
    }

    private function searchContacts():void
    {
        var dbStatement:SQLStatement = new SQLStatement();
        dbStatement.sqlConnection = dbConn;
        dbStatement.text = "SELECT * FROM contacts WHERE "+search
_cb.selectedItem+" LIKE :SEARCHVALUE";

        dbStatement.parameters[":SEARCHVALUE"] = search_txt
.text;

        dbStatement.addEventListener(SQLEvent.RESULT, search
ContactsResult);
        dbStatement.execute();
    }

    private function searchContactsResult(evt:SQLEvent):void
    {
        search_dg.dataProvider = evt.target.getResult().data;
    }
    ]}]>
</mx:Script>

<mx:TabNavigator left="0" right="0" top="0" bottom="0" id="tab_
nav">
<mx:Canvas label="List contacts" width="100%" height="100%">
<mx:DataGrid top="10" left="10" right="10" bottom="10" id=
"list_dg" />
</mx:Canvas>
<mx:Canvas label="Add contact" width="100%" height="100%">
<mx:Label x="10" y="10" text="Name"/>
<mx:Label x="10" y="40" text="Last name"/>
<mx:Label x="10" y="70" text="Email"/>
<mx:Button x="9" y="112" label="Add contact" click="add

```

Adobe AIR Architecture



```

Contact])"/>
<mx:TextInput x="93" y="8" id="name_txt"/>
<mx:TextInput x="93" y="38" id="lastname_txt"/>
<mx:TextInput x="93" y="68" id="email_txt"/>
</mx:Canvas>
<mx:Canvas label="Search contacts" width="100%" height="100%">
<mx:TextInput x="10" y="10" width="215" id="search_txt"/>
<mx:ComboBox x="233" y="10" dataProvider="["name",
'lastname', 'email']" id="search_cb"/>
<mx:Button x="338" y="10" label="Search" id="search_btn"
click="searchContacts"/>
<mx:DataGrid left="10" top="51" right="10" bottom="10" id
="search_dg"/>
</mx:Canvas>
</mx:TabNavigator>
</mx:WindowedApplication>

```

Ici la base de données de l'application est stockée en local dans le fichier "contacts.db". Deux objets sont à chaque fois nécessaires pour communiquer avec la base SQLite : *SQLConnection* et *SQLStatement*. A l'initialisation de l'application, on crée une table contacts si elle est inexistante sur le disque. L'objet privé *dbConn* reste persistant le temps d'exécution de l'application. Pour lister le contenu de la base, une fois la connexion établie (variable *dbConn* dans l'exemple), on crée une nouvelle requête avec l'objet *SQLStatement*. Dans la fonction *listContacts*, un simple 'SELECT * FROM contacts' suffit. Le retour d'un résultat est géré dans la fonction *listContactsResult* qui nourrit directement le dataProvider du dataGrid *list_dg*. Pour rajouter un contact ou effectuer une recherche, il suffit d'affiner sa requête et de récupérer des objets *SQLStatement*. La méthode *execute()* déclenche la requête à la base.

■ **Michaël CHAIZE** - Technical Sales - Flex & LiveCycle ES

Liens utiles

Blog de Michaël Chaze : <http://codemouinmouton.wordpress.com>
 Portail Flex : <http://flex.org>
 Tutoriaux et articles pour les développeurs :
http://www.adobe.com/fr/devnet/TogN-FR-gntray_comm_devnet_fr
 Portail sur la RIA : <http://www.riapedia.com>
 Communauté Flex française : <http://www.tfx.org>
 Portail sur Adobe et l'Open Source : <http://opensource.adobe.com>

L'INFORMATION du DÉCIDEUR

Choisir, déployer, exploiter les logiciels

Lisez le seul magazine offrant
aux responsables informatiques
une information et des témoignages
focalisés sur le logiciel en entreprise.

Egalement au sommaire du numéro :

- Lotus, au-delà de la messagerie
- Avis d'expert : ERP, génération 2.0
- Déploiement Vista : les outils d'administration
- La Virtualisation pour consolider ses serveurs
- Le tableau de bord de Windev 12
- Mener à bien son projet de développement
- Adopter la SOA : quels bénéfices pour l'entreprise ?



L'actualité
au quotidien :
• Sécurité • Projets
et développement
• Administration
• Progiiciels

Les Cas Clients

Prochainement : **Vidéos** (Actualité et Cas Clients)

www.solutions-logiciels.com

LE MAGAZINE DU DÉCIDEUR INFORMATIQUE EN ENTREPRISE
SOLUTIONS LOGICIELS N°1
FÉVRIER / MARS 2009
NOUVEAU
www.solutions-logiciels.com

55000 RECRUTEMENTS EN 2008
Les **PROFILS** les plus demandés
Comment les **SSII** attirent les talents

ERP
Un choix crucial

Les étapes
de la mise en œuvre

SÉCURITÉ

Boîtier **UTM**
arme absolue contre
les menaces internet

"GOOGLISER"
votre entreprise ?



Déployer Vista

✓ Faut-il passer à Vista ?
✓ Les pièges à éviter

TEMOIGNAGE
Déploiements
sur 130 postes

Alexandre POPESCU
IT manager, Dixon Wilson

N°1 - Février/Mars 2009
Chez votre marchand de journaux
Abonnez-vous (coupon ci-dessous)

OUI, je m'abonne (écrire en lettres capitales)

Envoyer par la poste à : Solutions-Logiciels, service Diffusion, 22 rue René Boulanger, 75472 PARIS - ou par fax : 01 55 56 70 20

1 an : 25€ au lieu de 40€, prix au numéro (Tarif France métropolitaine) - Autres destinations : CEE et Suisse : 30€ - Algérie, Maroc, Tunisie : 33€ - Canada : 39,50€ - Dom : 38€ Tom : 50€

8 numéros. Prochaines parutions : N°2 Avril/mai - N°3 Juin/Jul/août - N°4 Septembre - N°5 Octobre - N°6 Novembre - N°7 Décembre/Janvier - N°8 Février 2009 - N°9 Mars 2009

M. Mme Mlle Société

Titre : Fonction : Directeur informatique Responsable informatique Autre

NOM Prénom

N° rue

Complément

Code postal : Ville

Adresse mail

Je joins mon règlement par chèque à l'ordre de SOLUTIONS LOGICIELS Je souhaite régler à réception de facture

Sur la route de Java 7 :

Les promesses du changement

Java 7 sortira des laboratoires et de la communauté, d'ici un an environ. Nous savons déjà que cette future version reposera sur les nouvelles fondations de Java 6 Update N et du concept modulaire qui y a été introduit : JRE Consumer. De l'aveu même de Sun, Java reste trop lié à l'entreprise, et bien peu présent auprès du grand public. Pour se rendre attractif, il fallait réduire la taille de la JRE, améliorer les performances et le temps de démarrage. Un énorme travail dont on verra les premiers résultats dans la prochaine mise à jour de Java 6 qui devrait sortir à l'occasion de la JavaOne en mai prochain, ou quelques semaines après. Java 7 ne fera qu'étendre et améliorer cette profonde modification de l'architecture qui, on l'espère tous, impactera peu les applications actuelles.



A lors que Java 7 devait être la première version open source de notre café favori, Sun a décidé de changer de tactique ! Récemment, l'éditeur a publié dans le projet OpenJDK, le code source de Java 6 ! Il s'agit là d'un tournant important. Car si Java garde une omniprésence en entreprise, sur le développement web, mobile, RIA, RDA, les autres langages avancent rapidement. Microsoft, Adobe, Ruby, Mozilla, Google, n'attendent pas. Et c'est notamment le cas sur le mobile où Java, bien que présent, souffre d'une fragmentation importante, la promesse de Java FX Mobile de rationaliser tout cela tarde à se concrétiser. Espérons que la JavaOne, le 6 mai prochain à San Francisco, pourra lever les doutes (pareillement sur la partie desktop). L'arrivée prochaine de Microsoft avec Silverlight Mobile, le renforce-

ment de la partie logicielle de l'iPhone avec son SDK dédié, la disponibilité d'Android à partir de la fin de l'été, ou encore d'Adobe et de Mozilla, risquent de bouleverser le marché. Et Java a beau être présent sur des centaines de millions de terminaux, cela ne constitue plus un gage de succès ! C'est pourquoi la JVM se met aussi aux langages de script, aux langages dynamiques. Ce n'est pas un hasard si Sun mise beaucoup sur JRuby. Incontestablement, Java est aujourd'hui arrivé à une charnière de son existence et pas seulement parce qu'il est open source. La plate-forme a besoin de se simplifier, de nouveau, de rendre plus clair le modèle de développement, même si depuis Java 5, de gros efforts ont été réalisés. La modularité de la JRE aidera à améliorer le déploiement. Mais il lui faudra trouver sa place dans les der-

nières évolutions du développement web et desktop. Car les technologies de type Silverlight, WPF, Flex, Air, voire XUL, surclassent Java et occupent le terrain, un terrain qui ne sera pas évident à récupérer, car les éditeurs veulent agir rapidement.

Quel est l'avenir de NetBeans ? S'il a des qualités évidentes, l'omnipotence d'Eclipse gêne l'IDE à réellement s'épanouir sur le marché et bien peu d'éditeurs s'appuient dessus. Sun doit encore convaincre sur la partie outil, même si Glassfish a réussi à se faire une place.

Dans ce grand dossier, nous allons aborder les nouveautés de Java 6 Update N et vous dévoiler les principales nouveautés de Java 7 telles qu'on les connaît aujourd'hui. L'interopérabilité Java .Net sera aussi abordée ainsi que les derniers frameworks disponibles.

■ François Tonic

Les promesses de Java 7

La prochaine version de Java – nom de code "dolphin" – sera la septième selon la numérotation "produit" de SUN. Peu de choses ont été fixées pour cette nouvelle version. Il n'existe même pas de JSR la concernant ! Quand pourrons-nous en disposer ? Mystère ... Et pourtant, l'information serait très appréciée, car il semblerait bien qu'il s'agisse d'une "grosse" version, à l'instar de ce que fut, en 2005, le JDK 5.0 avec les annotations et les generics.

Mais voilà, tout doit être mis au conditionnel, et un usage intensif en est fait dans cet article. S'il est possible de glaner des informations sur les blogs des experts impliqués dans les diverses API qui pourraient être embarquées dans le JDK 7.0, il est impossible aujourd'hui de trouver LE document de référence qui indique ce que contiendra réellement cette version.

Comme le JDK 5.0, de notables évolutions de la syntaxe de Java seraient prévues. On trouve d'abord quelques ajustements mineurs facilitant ça et là l'écriture de code. Mais la principale nouveauté attendue est l'introduction des "closures" (appelées parfois "expressions lambda") dans le langage Java. Notons que les "closures" existent sur .NET à partir de la version 3.0.

Une autre nouveauté qui risque de faire date est la définition de modules Java (Java Module), sorte de super JAR (Java Archive), permettant de faciliter et fiabiliser la gestion des bibliothèques de classes.

Swing n'est pas oublié : SUN continue à croire au destin de Java sur le poste client, comme le montrait déjà l'initiative JavaFX. Le JDK 7 devrait embarquer les JSR 295 (Bean Bindings), JSR 296 (Swing Framework) et JSR 303 (Bean Validation). Un peu plus d'API et donc de complexité pour un mécanisme graphique déjà très riche (trop riche ?). Cela facilitera probablement l'écriture d'applications desktop, mais nécessitera un apprentissage supplémentaire. Beaucoup d'équipes de développeurs Swing risquent de se tourner vers des solutions globalement plus simples comme JavaFX. On peut certes objecter qu'un framework standard est de toute façon préférable à un framework "maison" bâti au dessus de Swing. Il existe bien d'autres projets qui pourraient (le conditionnel encore !) faire partie de Java 7. Une nouvelle version de NIO. Une nouvelle API pour gérer les dates (c'était vraiment trop simple avec les classes Date et Calendar). Une gestion standard des caches d'objets. Une gestion des unités et des quantités (JSR 275). Des apports à l'API de gestion de la concurrence, etc., etc.

Comme on ne peut pas parler de tous ces sujets – d'autant plus que beaucoup sont encore à l'état d'ébauches – je ne vais présenter en détail que deux des plus significatifs : les closures et les modules Java.

Les "closures"

Les "closures", fermetures en Français (aussi appelées "expressions lambda") ne sont pas une nouveauté dans les langages informatiques : des langages aussi divers que Smalltalk, JavaScript, C# ou Ruby en disposent déjà. Le tour de Java arrive. De quoi s'agit-il ? Il s'agit de définir, de manière légère, des fonctions capables de disposer du contexte d'un objet ou d'une autre fonction. Java dispose depuis longtemps, avec les classes anonymes, d'un mécanisme qui s'en rapproche. La version 7 propose donc une extension qui facilite encore davantage l'écriture de fonctions à la volée.



Une "closure" est un objet qui se définit de la façon suivante :

```
{int, int => int} fctAdd = {int x, int y => x+y};
```

La référence fctAdd désigne la closure. La déclaration (int, int => int) indique que fctAdd peut référencer une closure qui accepte en paramètre deux entiers et retourne un entier. L'objet closure est défini par {int x, int y => x+y}. Les paramètres en entrée sont nommés x et y. Leur somme est retournée par la closure.

La syntaxe d'une closure est la suivante :

```
{[paramètres] => [instructions][valeur retournée]};
```

Le code contenu dans une closure peut être arbitrairement complexe. La closure suivante retourne la valeur absolue d'une valeur passée en paramètre. Elle intègre pour cela des instructions devant l'expression finale :

```
{int=>int} fctAbs = {int x=>if (x<0)x=-x};
```

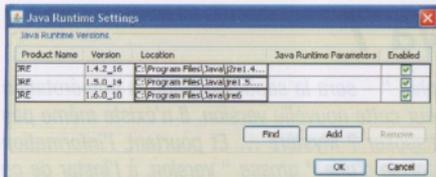
Où le "x" qui suit le ";" est l'expression qui est retournée par la closure. Elle peut très bien n'avoir ni paramètre, ni valeur de retour :

```
{=>} fctHello = {=>System.out.println("Hello world!");};
```

L'invocation d'une closure est réalisée par l'appel à la méthode "invoke" qui est définie pour toute closure. La signature de cette méthode invoke correspond aux paramètres et à la valeur de retour de la closure.

Ainsi on écrira :

```
int somme = fctAdd.invoke(3,4);  
int volAbs = fctAbs.invoke(-5);  
fctHello.invoke();
```



Le mécanisme prend toute sa saveur lorsque la closure utilise des attributs ou des variables (même locales !) accessibles à l'endroit où elle a été définie :

```
class Sprite {
    int x; int y;

    public void avancer() { agir(=>x+=1); }
    public void reculer() { agir(=>x-=1); }
    public void monter() { agir(=>y-=1); }
    public void descendre() { agir(=>y+=1); }

    void agir(=> action) {
        action.invoke();
        refreshDisplay();
    }
}
```

Cet exemple montre tout l'intérêt des closures : il devient possible de "passer" des instructions comme paramètres à une méthode. La chose était déjà possible en Java, mais de façon nettement plus lourde en implémentant à la volée l'interface Runnable :

```
public void avancer() {
    agir(new Runnable() {
        void run() { x += 1 ; }
    });
}

void agir(Runnable action) {
    action.run();
    refreshDisplay();
}
```

Bien sûr, l'utilisation de l'interface Runnable n'est possible ici que parce que la méthode d'action n'a besoin d'aucun paramètre, ni de valeur de retour. Si ce n'était pas le cas, il faudrait définir une interface ad hoc. C'est tout cela que nous épargne la closure. Il est même légal d'écrire :

```
class Sprite {
    {int=>} fctAvancer;
    {int=>} fctReculer;

    public void initActions() {
```

```
int x = (int)(Math.random()*100);
fctAvancer = {=>x+=1;};
fctReculer = {=>x-=1;};
}
}
```

Surprenant, n'est-ce pas ? Notre objet n'a plus d'attribut de positionnement. Celui-ci est défini comme variable locale d'une méthode et est ensuite manipulé par deux closures. Celles-ci peuvent être utilisées en dehors de la méthode initActions. Et pourtant la variable x est toujours disponible ! Cela n'est possible que parce que les variables locales manipulées par des closures sont allouées sur le tas (heap) et non sur la pile. Noter que nul spécificateur "final" n'est requis.

Le compilateur matérialise les closures en générant à la volée une interface et une classe implémentant cette interface. L'interface comme la classe ne définissent qu'une méthode : invoke, que nous avons déjà présentée.

Les closures sont bien sûr compatibles avec les autres mécanismes du langage comme les annotations ou les generics. Que veut-elles nous apporter ? Une nouvelle manière de programmer probablement. On va pouvoir, grâce à elles, accrocher facilement des comportements à des objets. C'est une façon très souple pour spécialiser le comportement d'un objet.

Les modules Java

Les développeurs Java qui se sont un tant soi peu frottés à des applications d'entreprise ont nécessairement connu le "JAR Hell", l'enfer des JAR : Les mêmes bibliothèques, utilisées par d'autres bibliothèques, dans des versions différentes, pas toujours compatibles ... Amis développeurs, votre souffrance a été entendue : SUN s'est décidé à introduire les modules Java sur sa plate-forme.

Un module Java est, comme un jar, une collection de classes compilées rassemblées dans un fichier zip, tout comme un fichier JAR. Il contient en plus une définition de module contenue dans un répertoire nommé MODULE-INF. Ce répertoire contient un fichier METADATA.module. Ce fichier contient des informations qui permettront à la JVM (Java Virtual Machine) de l'exploiter :

- Un nom de module.
- Sa version.
- La liste des modules dont il dépend avec éventuellement des contraintes sur les versions acceptables.
- La liste des points d'entrées publiques du module (les classes visibles en dehors du module).
- La classe d'amorçage du module (à l'instar de ce qui se fait dans les JAR). La fabrication d'un module Java est réalisée en écrivant un fichier "super package". Il s'agit d'un fichier java (super-package java) contenant les informations que l'on retrouvera dans le fichier METADATA.module. Ces informations seront aussi accessibles par introspection (une méthode getModule() sera disponible dans l'API de la classe ClassLoader). Un fichier super-package aura une structure qui devrait ressembler à cela (le conditionnel est de mise, la spécification n'a pas encore été validée) :

```
import java.module.annotation.*;

@MainClass("com.mypapp.mypkg.Launch") ;
```

```

@Version("1.1")
superpackage com.myapp.mypkg {

    @VersionConstraint("2.1+")
    import com.myapp.mybasepkg;

    @VersionConstraint("1.2*")
    import com.myapp.myotherpkg;

    member package
    com.myapp.mypkg
    com.myapp.mypkg.service
    com myapp.mypkg.gui

    export com.myapp.Launch;
    export com.myapp.Service;
}

```

Cette structure de fichier est nouvelle. Le mot clé superpackage introduit le nom du module (com.myapp.mypkg). Il peut être annoté pour indiquer sa version (1.1) et une éventuelle classe d'amorçage (com.myapp.mypkg.Launch).

A l'intérieur de la structure superpackage, on trouve des directives d'import. Elles indiquent quels sont les modules dont notre module a besoin. Ces directives peuvent être annotées afin de préciser quelles sont les versions acceptables de ces modules externes :

- Une contrainte de version de type X.Y+ indique que toute version égale ou postérieure à la version X.Y est acceptée
- Une contrainte de version de type X.Y* indique que toute version de type X.Y est acceptée (X.Y, X.Y.1, X.Y.2, etc.), mais pas une version dont le numéro X ou Y est incrémenté (pour une contrainte 1.2, une version 2.0 ou 1.3 ne sera pas acceptée).

La clause " member " indique quels sont les packages et les classes qui devront être embarqués dans le module. Les clauses " export " dressent la liste des classes qui seront accessibles depuis les autres modules. Un mécanisme de référentiel sera associé aux superpac-

kages. Il devra permettre d'enregistrer, retrouver et charger des modules, et ce avec un niveau de contrôle très supérieur au mécanisme actuel, basé sur les JAR et le CLASSPATH, qu'il devrait donc remplacer. On se demande, en voyant tout ce qui est annoncé, comment la compatibilité ascendante pourra être assurée. A priori, les modules Java devraient être compatibles avec les versions inférieures du JDK (avec des fonctionnalités dégradées probablement).

Que faut-il en penser ?

Le moins que l'on puisse dire est que Java 7 est encore dans les limbes. Les promesses semblent aussi riches que lointaines. D'autant plus que pour en profiter pleinement, il faudra que les IDE comme Eclipse, NetBeans ou IdeaJ en tiennent compte (évolution du compilateur et de l'analyseur syntaxique, prise en compte des nouvelles API graphiques dans les compositeurs graphiques, et ainsi de suite). Combien de temps faudra-t-il ? Plusieurs années sans doute ... Ce constat est un peu inquiétant car il montre un certain ralentissement dans l'évolution de la plate-forme Java. Est-ce la rançon de l'ouverture en " open source " ? Est-ce le destin d'une plate-forme déjà très complète qui repousse toujours plus loin les frontières du possible ? Est-ce une forme de prudence pour être davantage à l'écoute de la communauté des développeurs, dont le rôle décisif – démontré par les initiatives Hibernate ou Spring – ne peut plus être ignoré ? Il est aussi nécessaire que les développeurs Java " digèrent " les apports de la version 1.5 (et dans une moindre mesure 1.6). Nous avons dès maintenant les annotations et les generics. Mais leur utilisation reste encore assez limitée. La version 7 en tirera un meilleur profit. Combiné aux closures – un nouveauté de la version 7 – la programmation Java pourrait en être complètement bouleversée.



■ Henri Darmet

Directeur Technique - Objet Direct / Homsys Group
Objet Direct est une société spécialisée sur les technologies objet et Web : Conseil en méthodologie, en architecture et en urbanisation du SI, conception et mise en œuvre de projets, édition et distribution de logiciels, formation.
Objet Direct est une filiale de Homsys Group.
www.objetdirect.com

Programmez!

LE MAGAZINE DU DÉVELOPPEMENT

Prochain numéro
Hors Série .Net :
16 avril 2008

• Live 2.0 :

Découvrez la plate-forme des services en ligne Live, toutes les API, comment développer et implémenter les services Live.

• Parallel FX :

La programmation multicoeur " facile " !
Comprendre et utiliser une librairie surpuissante dans vos applications.

Java 7 : en passant par Java 6 Update N

La prochaine version de Java n'est pas encore sortie, mais de nombreuses propositions sont faites et apparaissent comme des fonctionnalités éligibles pour la sortie de Java 7.

La plate-forme de Sun est décrite par certains architectes de renom comme étant arrivée à mi-vie. Cet état de fait est principalement dû à la percée des langages de types RAD tels que Ruby ou encore à la concurrence du framework .NET de Microsoft. Il ne s'agit donc pas d'une simple mise à jour mais réellement d'une évolution du langage vers plus de simplicité, d'interopérabilité et de performances devant lui permettre de tenir à distance ses concurrents pour quelques années encore.

Plus de modularité

Tout d'abord, le langage devrait gagner en modularité avec notamment l'apparition des superpackages (JSR 294). L'idée de départ est que le mécanisme de package n'est pas suffisant pour structurer le code et qu'il est plus pratique de disposer d'ensemble de packages dont certains sont publics et d'autres internes, utilisés pour l'implémentation. C'est ce que permet la notion de superpackage. Ils seront surtout utilisés en phase de développement. Voici un exemple :

```
superpackage com.foo {  
  member package com.foo.api, com.foo.model; // Packages internes  
  export com.foo.api; // Package public  
}
```

Pour aller plus loin, l'arrivée d'un système de modules (JSR277) devrait améliorer significativement la modularité du langage en apportant une nouvelle architecture de déploiement. Celle-ci prendra en charge la gestion de version, les dépendances des modules et bien d'autres fonctionnalités encore. Tout cela sera géré à l'exécution, contrairement à l'outil de compilation Maven qui prend déjà en charge ces fonctionnalités. Concrètement il s'agira d'un format supplémentaire : le JAM. Celui-ci étend les fichiers JAR en y ajoutant des méta-données.

Pour conclure avec la modularité de Java, le JRE va subir une réorganisation majeure. En s'appuyant sur le Java Module System, Java sera désormais composé d'un kernel et de modules chargeable à la demande. L'objectif est de rivaliser avec les technologies Silverlight et Flex en diminuant de manière drastique le temps de chargement des applets et des applications Swing. Cela devrait également diminuer l'utilisation mémoire des applications Java.

JRE Consumer alias Java 6 Update N

Le JRE Consumer propose de corriger les défauts reprochés aux versions précédentes de Java : temps de démarrage, taille des applications et détection du java runtime. La majorité des fonctionnalités ont été intégrées dans la version 6 update N de la plate-forme mais le tra-

vail continue pour la version 7. Le JRE Consumer se constitue de plusieurs composants adressant chacun une fonctionnalité différente. Le Java Kernel dont nous venons de parler est l'une des fonctionnalités clef apportée par le JRE Consumer. Le Quickstarter en est une autre.

L'objectif du Quickstarter est de réduire le temps de démarrage des applications Java et plus particulièrement des Applets. Ce composant permet à Java de faire un pas en avant important en terme de performances. Cette modification du comportement de la machine virtuelle permet d'exploiter les caches des disques durs en préchargeant certains composants au démarrage du système. Cela a l'avantage d'annuler tout temps de chargement. En revanche, une partie de la mémoire est monopolisée.

La détection de la présence et de la version de Java est restée problématique jusqu'à présent. Avec l'arrivée de Java 5, un composant ActiveX permettait d'interagir avec Java, mais celui-ci se limitait à une utilisation d'Internet Explorer. La nécessité d'un mécanisme standard et largement supporté par l'ensemble des navigateurs s'est concrétisée par un outil de déploiement. Celui-ci permet à la fois la détection de Java depuis un plug-in et depuis du code JavaScript. Le plug-in reste cependant plus puissant et permet notamment de lancer automatiquement une mise à jour ou une installation avec un maximum de sécurité. Les installateurs de JRE et de SDK ont également évolué, ils vont apporter un gain en expérience utilisateur qui permet à la plate-forme de suivre la tendance actuelle et l'engouement pour les interfaces graphiques évoluées.

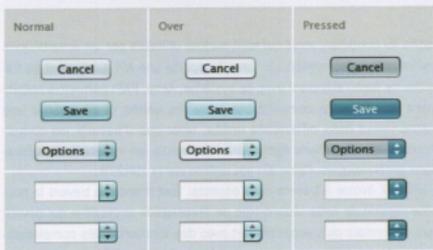
Les interfaces Swing des systèmes Mac OSX et Linux jouissaient de l'accélération graphique offerte par OpenGL. Désormais, les utilisateurs de Windows seront aussi bien lotis. Sun a réécrit le pipeline de rendu afin de permettre l'exploitation de l'accélération DirectX. Cela devrait permettre à toutes les plates-formes de disposer d'un niveau de performance équivalent. Les applications graphiques seront plus rapides et les développeurs pourront se permettre d'intégrer des composants graphiques évolués ainsi que des effets sans crainte.

Toujours au niveau des interfaces graphiques, le nouveau thème Nimbus fait son apparition et vient remplacer le vieillissant Metal. Ce dernier proposé par défaut a fait son temps et méritait de prendre sa retraite. Nimbus propose une interface mise au goût du jour et améliore la compatibilité entre les différentes plates-formes. Actuellement, Nimbus est implémenté par l'utilisation de Synth : un outil efficace pour créer des thèmes sans écrire de code Java. Le format XML permet la définition du style des applications.

Swing plus productif

Java 7 disposera d'un framework pour le développement d'applications Swing. Celui-ci devrait fournir un nombre conséquent de templates per-





mettant ainsi de couvrir les besoins classiques de la majorité des développements. Le développement Swing s'oriente donc vers une approche par composant bien plus productive. Des classes utilitaires permettront d'implémenter des mécanismes plus rapides pour la prise en charge de la persistance, de la gestion des ressources ou encore de la gestion des threads.

Lors du développement d'IHM, il y a souvent des widgets qui doivent posséder les mêmes valeurs. Imaginez par exemple un champ texte dont le contenu est la sélection d'une liste. Un nouveau framework du nom de Bean Binding Framework (JSR 295) permettra de lier directement deux composants afin de s'assurer que les contenus soient en permanence synchronisés. Voici un exemple :

```
Binding binding = new Binding(source, "sourcePath", target, "targetPath");
binding.bind();
```

Il est probable que des validateurs de bean soient intégrés au langage sous forme d'annotations à la manière d'Hibernate. Cette approche est très pragmatique et permet une validation efficace des formulaires.

Une machine virtuelle polyglotte

La nouvelle JVM devrait permettre l'exécution native des langages de script tels que Javascript, Beanshell, Groovy, Ruby ou PHP au sein de la machine virtuelle. Ce tour de force sera possible par la création d'un nouveau bytecode : `InvokedDynamic`. Celui-ci permet de répondre à la

problématique du typage faible de ces langages en apportant un mécanisme de sélection dynamique des méthodes à appeler. A l'heure actuelle, on parle souvent des bénéfices de Groovy ou de Ruby au sein d'une démarche commune : le développement rapide d'applications. Il faut faire attention ici car le support de ces langages ne se fait pas d'un coup de baguette magique et il faut avoir à l'esprit qu'actuellement le support de ces langages par les outils de développements reste encore limité. Les performances ne sont également pas encore au rendez-vous. Cependant, à choisir entre Rails et Groovy, ce dernier reste plus intéressant pour le développement d'applications d'entreprises et son intégration reste une bonne nouvelle.

Plus de souplesse pour les types

Les types génériques ont été introduits dans Java 5 et afin d'assurer la compatibilité avec le code existant ils ont été implémentés en écrasant les types. En d'autres termes, les types génériques ne sont pas distingués les uns des autres à l'exécution. Cela signifie que les classes `ArrayList<String>` et `ArrayList<Integer>` sont les mêmes. Dans de nombreux cas, cela ne pose pas de problème mais il est possible que cela provoque des effets de bord. Pour cette raison, Java 7 implémente la réification. Celle-ci permettra la récupération des informations sur les objets génériques et ainsi de rendre possible la distinction au moment de l'exécution entre types génériques. Une évolution du langage permettant de simplifier la déclaration des variables : l'opérateur de déclaration simplifiée. Celui-ci permet d'éliminer la redondance lors de la déclaration d'une variable. Son utilisation est encore plus justifiée pour les types génériques, comme le montre le code suivant

```
HashMap<String, Integer> map := new HashMap<String, Integer>();
```

peut être remplacée par la suivante :

```
map := new HashMap<String, Integer>;
```

Intégration du langage XML

A partir de java 7, il sera possible d'intégrer directement le code XML aux sources Java. Le support du format XML permettra également la construction de littérales, la conversion de données, la navigation ou encore le streaming. On pourrait par exemple ajouter un nœud à un élément de cette manière :

```
elt.appendChild{
  <muppet>
  <name>Kermit</name>
  </muppet>;
```

L'objectif est de fournir une classe aussi puissante que `String` en s'appuyant sur cette intégration du code XML au niveau du langage.

```
void rejectOpenFeatures(XML featureList) {
  List<XML> fs = featureList.findAll( "feature[state]='approved'");

  for(XML f : fs) {
    f.get("state").set("rejected");
  }
}
```

D'autres évolutions du langage

Java 7 risque de connaître l'arrivée des propriétés, une notion en provenance directement du framework .NET ainsi que des closures. D'autre part, les opérations seront également plus souples avec notamment la possibilité de réaliser des opérations sur les types `BigDecimal`. Les blocs `switch` pourront prendre en charge les `String` et non plus le trop limité type `int`. Enfin, les comparaisons entre types `enum` seront possibles.

Les évolutions de l'API

Java NIO2 va continuer le travail entamé par les ingénieurs de Sun avec la première version de NIO : fournir aux développeurs un ensemble d'API d'entrée / sortie optimisées. Les composants majeurs prévus pour Java 7 sont les suivants :

- Une nouvelle interface avec les systèmes de fichier qui supporte les accès "bulk" sur les attributs des fichiers, notification de modification, API spécifiques des systèmes de fichiers ainsi qu'une interface fournisseur de service pour les implémentations "pluggable" de systèmes de fichier
- Une API sur les entrées / sorties asynchrones pour les fichiers et les sockets
- Fonctionnalités supplémentaires pour le socket channel avec, par exemple, la prise en charge du multicast

L'API existante sur les dates et le temps sera revisitée. L'objectif est de créer un modèle plus compréhensif en remplaçant les classes `Date` et `Calendar` par une nouvelle implémentation. Cette nouvelle implémentation devrait permettre de représenter les dates, durée et intervalles selon un format plus concis. Cela permettra d'augmenter la qualité du code. Par exemple, il ne sera plus nécessaire de recourir au type `int` pour stocker une durée avec un commentaire `javadoc` pour le décrire comme un nombre de jours. Un objet tiendra ce rôle.

Le système international définit les unités et les quantités que nous uti-



Les nouveautés Java vont vers la simplification

Ce que demandent les développeurs Java - et ce sur quoi s'orientent désormais les fournisseurs d'outils ou d'API - est de faire plus facilement les choses et non de faire de plus en plus de choses. L'adoption de JPA ou de JAX WS en est le signe tangible pour ce qui est des API standard. Les IDE offrent de plus en plus de fonctions d'automatisation de certaines tâches (wizards). Ils offrent enfin des compositeurs graphiques décents. Ils prennent mieux en compte les outils tiers (comme les serveurs d'application), à l'instar de WTP sur la dernière version d'Eclipse. Les frameworks qui prennent de l'importance, comme Seam, visent essentiellement à réduire l'effort du développeur. Les serveurs d'applications qui s'imposent sont d'abord ceux qui sont les plus simples à manipuler.

■ **Henri Darnet**, directeur technique (Object Direct)



Certains CV ne restent que 24h

" Concernant le marché de l'emploi Java en France, les entreprises doivent aujourd'hui faire face à une pénurie de compétences. Le recrutement de ce type de profil devient un véritable challenge pour les SSII, comme Valtech Technology. Dans la recherche de candidats, il faut aller très vite. Certains CV ne restent que 24h sur les CVthèques et les candidats peuvent recevoir jusqu'à 50 demandes par jour. Il faut également être très réactif dans les propositions de contrat et en parfaite cohérence avec les attentes des candidats qui ont généralement plusieurs offres en même temps. "

■ **Virginie Urtado** - Responsable Recrutement et Coordinatrice RH Valtech Technology Paris
Virginie.urtado@valtech.fr - www.valtech.fr

lisons au quotidien, mais qui sont surtout utilisées par les scientifiques (mètre, seconde, newton...). Java 7 apporte une API qui permet de travailler de façon standard avec ce système. Ainsi, l'API permettra notamment de vérifier la compatibilité entre les unités et d'exprimer une même valeur dans différentes unités, ainsi que d'assurer les conversions. L'API simplifiera également les calculs en intégrant les opérations arithmétiques. Cette API est définie par la JSR275.

L'API JMX évolue à travers deux aspects. Tout d'abord à travers la nouvelle version de cette technologie de messagerie applicative qui devrait être désormais plus accessible et avec des fonctionnalités complémentaires comme le support des annotations. Ensuite, JMX disposera d'un connecteur permettant de dialoguer avec une application Java sans pour autant que le client soit développé dans ce langage. Cette technique reposant sur les services web augmente l'interopérabilité du langage.

L'outil Javadoc

L'outil Javadoc sera revu notamment afin d'améliorer l'ergonomie de la documentation HTML générée. D'autres informations pourront également être ajoutées et l'intégration d'exemples au sein de la Javadoc sera mieux gérée avec un balisage spécifique. De la même manière, de nouvelles catégories verront également le jour. Toutes ces améliorations bénéficieront pleinement à la documentation des projets.

Conclusion

Java s'oriente vers plus de facilité, de souplesse et de modularité. L'adoption de langages de script au niveau de la machine virtuelle permet à Sun de conserver ses développeurs Java qui partent vers d'autres technologies de type RAD, dont le célèbre Ruby. Java 7 offre une grande bouffée d'air frais aux développeurs d'application de type client lourd et offre les avancées nécessaires à la technologie Java FX. Attendons 2009.

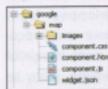
■ **Loïc Guillois**

Les nouveaux frameworks du monde Java



Phobos, Jmaki, ... L'écosystème Java s'enrichit aujourd'hui de nouveaux frameworks visant à faciliter le développement d'applications et gagner en productivité. Découvrons à travers cet article leurs fonctionnalités et l'intérêt de leur utilisation au quotidien.

Utilisée pour produire des sites web modernes et interactifs, la technologie Ajax est aujourd'hui omniprésente. Malheureusement, son intégration n'est pas toujours très simple: problème de compatibilité entre les différents navigateurs, protocole d'échange laborieux à mettre en place, etc. Afin de pallier ces difficultés, la communauté open source met à la disposition du développeur un certain nombre de frameworks, tous plus différents les uns des autres tant dans leur approche que dans leurs fonctionnalités. Parmi les plus récents, jMaki et DWR semblent sortir leur épingle du jeu.



jMaki

Issu des laboratoires Sun, jMaki est une large bibliothèque de composants graphiques (ou widgets) basés sur la technologie Ajax. Son modèle, conforme au format JSON, permet de regrouper dans un ensemble

homogène les widgets issus d'autres bibliothèques telles que Dojo, Scriptaculous, Yahoo, Google gears, etc. En outre, jMaki bénéficie de plug-in pour les environnements Eclipse et Netbeans, ce qui simplifie d'autant son utilisation : il suffit de glisser/déposer le composant sur la page et l'ensemble des ressources est instantanément copié dans le répertoire approprié. L'application peut s'exécuter sans autre paramètre.

```
<div id="content" style="height:400px">

  <a:widget name="google.map"
    args="{ centerLat : 37.4041960114344,
            centerLon : -122.008194923401 }" />

</div> <!-- content -->
```



Comme l'illustre la figure, chaque widget est défini par un style CSS, un gabarit HTML et un contrôleur Javascript. Cette structure garantit une adaptation rapide du widget aux spécificités de l'application. Côté serveur, le runtime de jMaki génère les identifiants nécessaires à certains composants et produit le code HTML envoyé au client. Autres avantages, le framework permet de récupérer des flux RSS ou encore d'accéder à des web services externes au travers de son module XmlHttpProxy.

page Javascript du client. Basée sur la technologie Ajax, la communication s'effectue de manière asynchrone, garantissant ainsi un fonctionnement fluide de l'application. Depuis la version 2.0 du framework, l'échange en sens inverse est également possible :



DWR

la méthode Java peut interagir avec une fonction Javascript, c'est le principe du Reverse Ajax. Utilisé pour enrichir une application web en préservant le code métier développé en Java, DWR constitue l'un des frameworks Ajax les plus prisés du moment. Son intégration s'effectue très simplement : dans un premier temps, il vous faut copier l'archive `dwr.jar` dans le répertoire `WEB-INF/lib` et déclarer la servlet principale dans le fichier de configuration `WEB-INF/web.xml`. Il suffit ensuite de spécifier la classe Java qui sera mappée au code Javascript :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dwr PUBLIC
  "-//GetAhead Limited//DTD Direct Web Remoting 2.0//EN"
  "http://getahead.org/dwr/dwr20.dtd">

<dwr>
<allow>
  <create creator="new" javascript="Demo">
  <param name="class" value="org.getahead.dwrdemo.simpletext.Demo"/>
  </create>
</allow>
</dwr>
```

La classe Java utilisée dans l'exemple retourne une simple chaîne de caractère :

```
package org.getahead.dwrdemo.simpletext;

public class Demo {
  public String sayHello(String name) {
    return "Hello, " + name;
  }
}
```

Côté client, le lien s'effectue très facilement :

```
function update() {
  var name = dwr.util.getValue("demoName");
  Demo.sayHello(name, function(data) {
    dwr.util.setValue("demoReply", data);
  });
}

<p>
```

DWR

Le framework DWR - Direct Web Remoting - disponible sous licence Apache permet d'appeler une méthode Java directement depuis une

Name:

```
<input type="text" id="demoName"/>
<input value="Send" type="button" onclick="update[]"/>
<br/>
Reply: <span id="demoReply"></span>
</p>
```

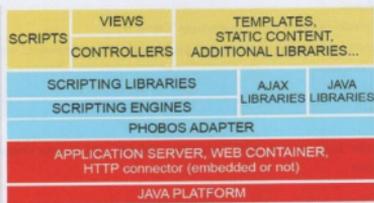
Gagner en souplesse avec les langages de script

Le développement rapide d'application (RAD) repose aujourd'hui sur une utilisation toujours plus importante de langages de script comme en témoigne le succès rencontré par les langages PHP, Ruby, Python. Fort de ce constat, Java implémente depuis le kit J2SE 6.0 les spécifications JSR 223 qui offrent aux développeurs une solution pour programmer dans différents langages de script :

```
ScriptEngineManager mgr = new ScriptEngineManager();
ScriptEngine jsEngine = mgr.getEngineByName("JavaScript");
try {
    jsEngine.eval("print('Hello, world!');");
} catch (ScriptException ex) {
    ex.printStackTrace();
}
```

La communauté open source, profitant de cette extension, a produit des langages de script de grande qualité fonctionnant sur la plate-forme Java. Voyons de plus près quelques uns de ces langages.

Phobos



Reléguant Java comme simple runtime, Phobos propose d'exécuter l'application scriptée directement sur le serveur. L'objectif est de simplifier le développement d'applications Web en occultant les difficultés liées à la compilation et au déploiement tout en conservant la fiabilité de la plate-forme Java. A l'heure actuelle, Phobos utilise le langage Javascript lequel, en plus d'adresser une très large communauté de développeurs, garantit une certaine cohérence du code en termes de nomenclature et de style. Néanmoins, l'objectif est de proposer à terme un support pour les autres langages phares du développement scripté à l'instar de Ruby. Autre qualité du framework, sa structure s'inspire du motif de conception MVC. Ainsi, une application Phobos est composée de :

- un contrôleur
- une vue
- un script pour dispatcher les différentes requêtes http.

Notons enfin que Phobos est habituellement associé au framework jMaki pour l'intégration de la technologie Ajax et l'utilisation de sa large bibliothèque de composants graphiques.

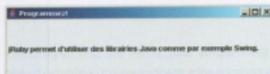
jRuby

jRuby est un interpréteur Ruby développé en Java. Bénéficiant de cette mixité de langages, il offre la souplesse du langage Ruby et garantit la pérennité du code métier déjà développé en Java. Voici une illustration de cette possibilité :

```
include Java

frame = javax.swing.JFrame.new("Programmez!")
label = javax.swing.JLabel.new("jRuby permet d'utiliser des librairies
Java comme par exemple Swing.")

frame.getContentPane.add(label)
frame.setDefaultCloseOperation(javax.swing.JFrame::EXIT_ON_CLOSE)
frame.pack
frame.setVisible(true)
```



Outre cette spécificité, le développeur peut opter pour une exécution en mode interprété

ou en mode compilé JIT (Just In Time). En mode interprété, jRuby se montre moins vélocité que l'interpréteur écrit en C. En revanche, en mode compilé, l'application gagne en performance, ce qui représente un atout majeur dans le choix de la solution. De plus, l'application peut s'exécuter sur les serveurs J2EE pour lesquels la fiabilité en environnement de production est largement reconnue aujourd'hui. En contre-partie, l'interpréteur jRuby dispose d'un choix plus restreint de librairies comparé à l'interpréteur natif écrit en C. Mais la communauté jRuby est très active et chaque jour de nouvelles librairies sont proposées sur Internet.

Grails

Très fortement inspiré de Ruby On Rails dont il tire en partie le nom, Grails est un framework orienté web et basé sur le langage Groovy. Son objectif est de fournir au développeur un cadre de travail regroupant l'ensemble des fonctionnalités essentielles à la réalisation d'une application web moderne. C'est ainsi qu'on retrouve au sein de Grails les frameworks Hibernate et Spring largement utilisés dans le monde Java et reconnus pour leur efficacité et leur fiabilité. En outre, l'utilisation du langage Groovy, proche de Java, permet une appréhension rapide du framework. Tout est conçu pour que le développement d'application soit le plus simple et le plus efficace possible. Et l'ensemble n'en reste pas moins flexible puisqu'il offre par exemple la possibilité de remplacer le module de présentation fourni initialement, par un framework plus spécifique tel que GWT.

Vous pourrez trouver plus de renseignements sur ces différents frameworks aux adresses suivantes :

jMaki - <https://ajax.dev.java.net>
DWR - <http://getahead.org/dwr>
Phobos - <https://phobos.dev.java.net>
jRuby - <http://jruby.codehaus.org>
Grails - <http://grails.org>

■ Vincent Perdereau - Chef de projet / ads-COM
vincent.perdereau@ads-com.fr - <http://www.ads-com.fr>

Pour les **.Net addicted**

Hors-Série **.Net**

Pro
Fevrier / Mars 2008 - Hors-Série

www.programmez.com

grammez!
LE MAGAZINE DU DÉVELOPPEMENT

.Net la révolution 2008 Spécial TechDays

WPF

Des interfaces
riches en WPF

Windows

Maîtriser les
fenêtres Vista !

Application

Développement par
composants en .Net

Multi-targeting :
la killer fonction !

Découvrez
ASP.Net MVC

Le nouveau visage
d'ASP.NET

.Net 3.0 / 3.5

Bien démarrer avec
Windows Workflow Foundation

Visual Studio Shell

Créer sa distribution
Visual Studio

M 02104 - 10 H - F: 4,95 € - RD



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

Demandez-le à votre marchand de journaux

Quelle interopérabilité Java / .NET ?

Définie de manière scolaire, l'interopérabilité d'un système est la capacité de celui-ci à fonctionner avec d'autres systèmes existants ou futurs. Ainsi, ses interfaces avec le monde extérieur doivent être connues et clairement définies.

Le sujet étant complexe, l'objet de cet article n'est pas de traiter l'ensemble des aspects de l'interopérabilité. Ici, nous allons présenter les différentes techniques et outils qui permettent à des applications Java et .NET d'interagir avec une compatibilité maximale.

Pour réaliser ce grand défi de l'interopérabilité entre Java et .NET deux grands noms de l'informatique collaborent : Sun Microsystems et Microsoft. Ce dernier a récemment annoncé des engagements d'ouverture qui agitent beaucoup les esprits. Microsoft affirme notamment sa volonté d'assurer des connexions ouvertes et de promouvoir la portabilité des données. Il affirme également vouloir s'engager dans l'open source. En ce qui concerne l'interopérabilité avec Java, le chantier est déjà bien avancé et de nombreuses ressources sont d'ores et déjà disponibles.

De son côté, Sun continue à travailler sur l'interopérabilité comme il l'a fait depuis de nombreuses années. Cependant ce n'est que récemment que Microsoft et Sun ont collaboré pour une interopérabilité entre le framework .NET dans sa version 3.0 et la pile JEE. Les développeurs pourront désormais tirer profit de nouvelles fonctionnalités au sein de l'environnement Netbeans et du serveur d'application Glassfish.

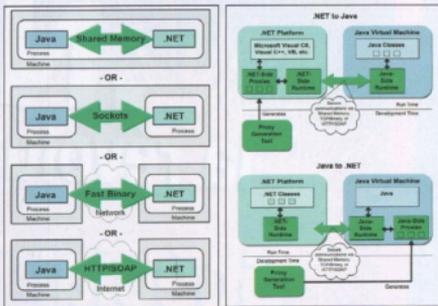
Encourager l'interopérabilité

Afin de favoriser le développement de l'interopérabilité au sein des systèmes d'informations, deux organisations travaillent sur la mise en œuvre de normes et de standards. Il s'agit du consortium W3C, bien connu pour sa présence sur le web, et de l'organisation OASIS, père du format OpenDocument et autres annuaires UDDI. L'ietf travaille également sur la problématique d'interopérabilité bien que cela ne corresponde pas au cœur de son activité.

Qu'il s'agisse de la solution OpenOffice soutenue par Sun ou de Microsoft Office, les deux environnements supportent de mieux en mieux les formats standard et interagissent de plus en plus, notamment à travers la technologie de service web comme le fait Excel. L'ensemble des API permettant de travailler sur ces formats est disponible en .NET et en Java. La problématique d'interopérabilité Java / .NET reste avant tout une problématique de développeur.

Publier l'applicatif et architecture SOA

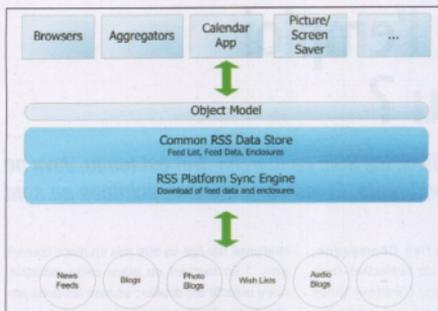
La tendance actuelle des développements d'applications d'entreprise est résolument orientée vers le web service. De nombreuses technologies sont utilisées pour publier l'applicatif : qu'il s'agisse de .NET ou Java, la volonté de décloisonner les systèmes a permis la maturation des architectures orientées services. L'approche naturelle pour l'interopérabilité .NET et Java est donc de s'appuyer sur ces solutions existantes utilisant notamment SOAP pour les services web purs et RSS pour la syndication de contenu métier. Tandis que les services web permettent une communication synchrone héritant des protocoles tels que CORBA, la syndication de contenu permet un découplage plus fort : une application métier continue son exécution sans attendre une autre.



L'outil le plus efficace dans le domaine des services web reste Axis qui permet la génération de code à partir des fichiers WSDL et inversement. A l'origine, ce projet adressait la plate-forme JEE, mais plus récemment Axis permet une intégration à la plate-forme .NET. Son utilisation automatise de nombreuses tâches et permet notamment la mise en œuvre simplifiée d'un conteneur de services web. Microsoft a poussé la technologie SOAP sur son serveur de bases de données SQL Server pour permettre l'interopérabilité. Ainsi, il est possible de profiter de l'ensemble de ses fonctionnalités qui vont désormais bien plus loin que le simple SQL depuis le monde Java. SQL Server propose en effet un broker de message, un moteur OLAP pour le décisionnel ou encore un ETL. L'utilisation de la syndication de contenu avec RSS est une nouvelle approche pour la publication de données métiers. Elle apporte une souplesse et une accessibilité plus importante au détriment de la sécurité. Elle reste cependant une approche efficace pour les données publiques. Internet Explorer 7 et le système d'exploitation Vista apportent une plate-forme complète pour consommer ce type de données. L'API permettant l'interfaçage avec cette plate-forme RSS est accessible .NET et est exposée au monde extérieur via la technologie COM. Celle-ci pourra donc être exploitée depuis Java par l'utilisation de connecteurs tiers et ces flux RSS pourront être générés de manière standard.

Technologies d'intégration et d'interopérabilité

JMS est une technologie de messagerie applicative. Elle permet l'envoi de messages synchrones et asynchrones. Elle est souvent utilisée comme support pour l'intégration d'applications et de web services au sein de middleware. Biztalk est un serveur de BPM qui assure pour prin-



cipale tâche l'orchestration de l'applcatif ainsi que le déroulement de processus métier. Biztalk supporte la technologie JMS, ce qui constitue un atout considérable pour l'interopérabilité et plus particulièrement dans le domaine des BPM, où .NET et Java peuvent être sollicités.

Biztalk est une offre propriétaire. Il existe cependant un outil qui réalise des fonctionnalités tout aussi intéressantes dans l'open source, et qui sur certains point va plus loin : ActiveMQ. Il s'agit d'un broker de message qui apporte également des fonctionnalités d'intégration. Il a été développé en Java pour des applications Java mais l'API de messagerie NMS permet aux applications .NET de l'utiliser avec une interopérabilité maximum. Le code source suivant montre comment envoyer un message applicatif au serveur.

```
using System;
using System.Collections.Generic;
using System.Text;

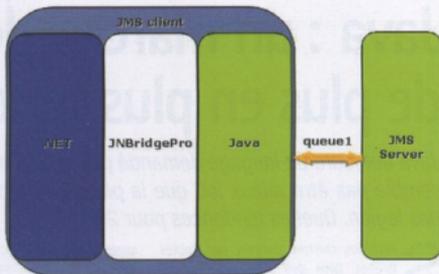
using ActiveMQ;
using Spring.Messaging.Nms;

namespace SenderConsole
{
    class Program
    {
        private const string URI = "tcp://localhost:61616";
        private const string DESTINATION = "test.queue";

        static void Main(string[] args)
        {
            ConnectionFactory connectionFactory = new ConnectionFactory(URI);
            NmsTemplate template = new NmsTemplate(connectionFactory);
            template.ConvertAndSend(DESTINATION, "Hello from the sender.");
        }
    }
}
```

Proxy et pont applicatif

Un certain nombre d'éditeurs proposent des API permettant la création de ponts entre applications Java et .NET. Les solutions JNBridge et J-Integra sont les plus avancées dans ce domaine et offrent des perspectives de premier ordre.



JNBridge permet la création de pont entre applications .NET et Java ainsi qu'entre le protocole de messagerie application JMS et le serveur BizTalk de Microsoft. JNBridge peut être utilisé pour créer des ponts bidirectionnels entre des applications dont on ne possède pas le code. L'outil génère des proxy permettant de publier les API des deux langages, ensuite il permet le choix de différentes techniques : mémoires partagées, sockets, web services... Le tout se construit grâce à une interface graphique, sans être confronté à des problématiques de programmation. JNBridge les prend en charge pour vous : ramasse miettes, conversion de types de données, cycle de vie des objets, gestion des exceptions... J-Integra est une solution qui permet d'aller plus loin dans le code en appelant par exemple les fonctionnalités d'une DLL .NET depuis une application Java. J-Integra est proposée dans différentes versions permettant la prise en charge des technologies de communication Corba, COM, Exchange, RMI et EJB. Contrairement à JNBridge, J-Integra est une API qui permet au développeur de réaliser lui-même ses ponts.

Le projet Mono

Mono est une solution libre permettant l'exécution d'applications .NET sur les plates-formes autres que Windows et notamment Linux. Mono est capable d'exécuter du code Java afin que les objets de ces deux plates-formes puissent interopérer. Ce tour de force a pu être réalisé par l'utilisation d'une machine virtuelle java au-dessus de .NET : IKVM. Actuellement, l'outil permet d'exécuter des applications populaires comme l'environnement Eclipse ou encore les serveur Derby et JBoss. IKVM peut être utilisé de deux manières. Avec la première méthode, on l'utilise comme un compilateur Just-in-Time qui va traduire le bytecode Java en binaire .NET à l'exécution. Cela provoque une perte de performance et pour cette raison la seconde méthode est privilégiée : la compilation Ahead-of-Time. Dans ce cas, le bytecode Java est transformé en un exécutable x86 pouvant être exécuté ultérieurement par Mono.

Perspectives

Les mondes Java et .NET se sont affrontés pendant des années. Cependant, pour le bon fonctionnement du système d'information, l'interopérabilité entre les plates-formes est devenue une nécessité. Les technologies sont présentes, mais souvent mal connues et peu utilisées. A l'avenir l'interopérabilité sera un défi majeur, notamment avec l'émergence de Linux au sein des entreprises et l'arrivée de Vista. Il y a fort à parier que de nombreux standards ouverts vont apparaître, sûrement par standardisation des technologies existantes.

■ Loïc Guillois

Java : un marché de l'emploi de plus en plus tendu ?

Java demeure un langage demandé par les entreprises, les SSII. Le marché .Net est tendu, Java ne semble pas être mieux loti que la plate-forme concurrente. Et les compétences pointues ne sont pas légion. Quelles tendances pour 2008 ?

Selon les chiffres donnés par Rachel Brunet, DRH des Joudis, les offres Java représentent un quart des annonces (soit env. 930 sur 5200). Ces demandes concernent l'ensemble du marché Java : développeurs, chef de projets, ingénieurs, architectes, etc. Java est un langage qui connaît une tendance à la hausse et cela ne semble pas se démentir. Entre 2004 et 2008, c'est 50 % en plus, indique-t-elle.

La demande est supérieure à l'offre

Le marché apparaît tendu. L'entreprise, la SSII ne cherche plus la perle rare mais des compétences. " Il y a plus d'offres que de demandes " poursuit Rachel Brunet. Ce n'est pas un hasard si dans certaines écoles, les entreprises cherchent " à la source " des compétences, et souvent avant que le diplôme ne soit délivré, manière de sécuriser une partie de son recrutement. Le turn over n'est bien entendu pas absent, mais celui-ci est difficile à évaluer. Avec un marché tendu, l'entreprise n'a pas intérêt à laisser partir ses compétences vers un concurrent potentiel. La question des salaires est devenue un enjeu. Le plus en plus souvent, le recruteur indique seulement " salaire à négocier ". " On ne veut pas fermer la porte " précise Rachel Brunet. Le salaire serait-il un levier pour le recruteur et le candidat ? Oui, et il le sera d'autant plus que les compétences et expertises seront élevées.

Des compétences plus rares en France ?

Un des soucis est de trouver des compétences adaptées. Or, aujourd'hui, la technique souffre d'une mauvaise image en France, on fait moins souvent carrière en étant " simple " développeur. Et beaucoup d'étudiants s'orientent vers des carrières de chef de projet, d'architecte. Cela n'est pas propre au monde Java. " Il est assez difficile de trouver la bonne personne. Le métier est ressenti comme trop dur. Ils font un séjour " rapide " (dans la technique)", analyse



Christian Champagne, PDG de Nettecive Technology. L'éditeur lance d'ailleurs une campagne de recrutement d'une quarantaine de personnes expertes en UML/ Java principalement. Et si l'édite

teur ne trouve pas en France, il ira les chercher en Europe, voire, en dehors.

Pour Franck Priore, directeur des Opérations Objet Direct, " Les bons profils de concepteurs développeurs Java, 3 à 7 ans d'expérience, sont devenus objectivement rares sur le marché. Le positionnement d'Objet Direct, spécialiste des architectures Java depuis plus de 10 ans, nous permet heureusement d'être très attractifs pour cette population : la densité de nos compétences, la richesse de notre offre de formation interne, l'intérêt technologique des missions proposées, nos outils de capitalisation (cf. Espace Techno - wiki sur www.objetdirect.com) garantissent à nos consultants un environnement de très grande qualité. " Il faut aussi sensibiliser l'étudiant. Nettecive, comme d'autres éditeurs, organise des concours, met sur pied des partenariats. Un

challenge Blu Age va être mis en place courant avril. " En mettant en place cette initiative, notre objectif est double : séduire les bons profils et identifier les étudiants les plus à même de devenir de futurs collaborateurs " déclare Christian Champagne. " Au cours de ce challenge, nous souhaitons permettre aux étudiants de se confronter à une méthode de développement originale, de vivre les défis technologiques passionnants qui sont notre quotidien, leur communiquer notre enthousiasme et au total leur donner envie de nous rejoindre ".

Se former encore et toujours

Un développeur Java, ou sur tout autre langage, doit toujours se former (s'auto-former bien souvent), suivre les évolutions, faire de la veille. Il est important, pour ne pas dire vital, de coller aux technologies. Le monde Java bouge beaucoup depuis quelques mois. Et il est toujours utile que le développeur puisse comprendre l'intérêt de tel framework, outil, fonction. Même si cela prend du temps, être à niveau, peut aussi vous aider à évoluer dans votre métier ou tout simplement vous permettre de changer de poste...

■ François Tonic



Eurogiciel : une quarantaine de postes à pourvoir

" Il y a une pénurie sur le profil expert avec 4/5 ans d'expérience. Par contre il n'y en a pas sur les profils débutants " prévient d'emblée Nicolas Petitjean, responsable du recrutement de la société Eurogiciel. Car si aujourd'hui les projets Java sont abondants, les problématiques se sont déplacées vers l'architecture, la SOA, les services web, le mapping, " ce sont des profils peu fréquents (sur le marché) " prévient M. Petitjean.

Et les gens en place le savent parfaitement ! Le marché est donc tendu. " Cela implique pour nous d'identifier les bons profils. Il faut savoir les capter ". Ce captage passe par plusieurs éléments : les projets, les perspectives, et bien entendu le salaire, même si sur ce dernier point, Nicolas Petitjean indique que cela n'arrive qu'en 3e position, derrière l'évolution du poste, les projets. Quant à la fourchette salariale, le responsable l'estime de 40 à 50 000 euros bruts (annuels). Paradoxalement, en 2006-2007, les écoles d'ingénieurs avaient 101 000 étudiants contre 37 000 en 2003-2004, mais beaucoup ne font pas carrière dans l'informatique et par conséquent dans le développement. Face à ce marché, " nous avons une quarantaine de postes à pourvoir actuellement ", précise M. Petitjean. C'est pour cela que la société forme des ingénieurs non informaticiens aux technologies Java (et autres).

Android : mettez de l'Open Source dans votre téléphone

Google vient de rendre public le SDK d'Android, un environnement de développement d'applications pour les Smartphones, basé sur des logiciels libres. La réalisation d'une première application est l'occasion d'évaluer ce nouveau système.

Attention : le SDK est toujours en pré-version. Il est mis à jour régulièrement. Les premiers téléphones Android ne seront pas disponibles avant le 3e trimestre.

Les logiciels pour les téléphones mobiles haut de gamme sont dominés par les solutions propriétaires: Apple, Symbian, Windows Mobile, Palm, ... Tous ces environnements se révèlent assez éloignés des environnements de développement standard (excepté Windows Mobile sur lequel on peut passer par .Net, ndr), et sont réservés à des développeurs spécialisés. Il existe, bien sûr, le mobile en environnement Java : J2ME, mais celui-ci est à la fois très allégué par rapport à J2SE, et fournit un accès limité aux fonctionnalités du téléphone.

Google, et le consortium handset, arrivent sur ce marché avec la volonté de changer la donne en proposant un environnement de développement ouvert : Android, basé sur les standards du marché et les logiciels libres. Voici un tour d'horizon des fonctionnalités du système, vu sous l'angle pratique de la réalisation d'une application.

Un système Linux sur le mobile

Le cœur d'Android est le système d'exploitation Linux 2.6. C'est une version pour les processeurs embarqués qui équipent les téléphones portables (la version de Linux livrée avec le SDK Android est une version pour processeur ARM). Linux possède des drivers adaptés au téléphone mobile comme la radio, l'appareil photo ou le Bluetooth et un système de fichiers "yaffs2" adapté aux mémoires flash.

L'utilisation de Linux est un gage de portabilité et de capacité à intégrer de nouveaux périphériques. Le portage d'Android sur un nouveau matériel va utiliser les outils de l'environnement de développement GNU avec le compilateur GCC et le debugger GDB, ainsi que l'ensemble des outils de mise au point du noyau pour la réalisation des drivers.

Linux n'est pas une nouveauté sur les Smartphones, Motorola propose déjà plusieurs modèles basés sur ce système d'exploitation. Mais Google innove en limitant Linux aux aspects systèmes, les applications devant être réalisées en Java. Android dispose pour cela d'une machine virtuelle spécifique : Dalvik.

Un environnement Java 5

L'environnement Java d'Android est une bonne surprise, finies les limitations de J2ME avec des profils limités enlevant tout espoir de portabilité des applications Java J2SE. Ce sont bien les principaux packages J2SE qui sont disponibles sous Android :

- java.lang
- java.security
- java.sql

- java.reflection
- java.io
- java.net
- java.nio
- ...

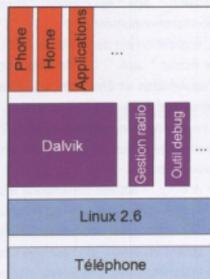
Les packages liés aux interfaces homme/machine (Swing ou AWT), à la communication inter-procesus (java.rmi) sont absents car ceux-ci sont remplacés par des composants spécifiques à Android. Les packages "java.*" disponibles sont, en apparence, très semblables à ceux de Java 5, mais on ne retrouve pas la documentation habituelle. En effet, pour s'affranchir des contraintes liées à la propriété intellectuelle de SUN, Google n'a pas utilisé le code de SUN, mais Apache Harmony qui est une implémentation open source de Java SE.L'usage montrera quel impact aura en terme de portabilité l'utilisation d'un autre run time que celui de Java d'origine. Mais l'aspect pratique de disposer d'un runtime Java 5 à ce niveau de compatibilité est incontestablement un point fort de la plate-forme.

- commons.httpclient d'Apache, librairie de référence pour permettre l'accès HTTP depuis Java.
- commons.codec d'Apache, librairie d'encodage des caractères.
- Webkit, librairie à la base de nombreux navigateurs comme ceux des séries S60 de Nokia ou de Safari. Elle possède l'avantage de disposer d'un moteur Javascript et d'être peu consommatrice de ressources. Sur Android, les fenêtres Webkit peuvent être entièrement générées depuis Java.
- OpenGL ES de Khronos, librairie 3D que l'on trouve déjà sur de nombreux téléphones mobiles. C'est une version simplifiée de la spécification OpenGL 1.5 adaptée aux processeurs graphiques des téléphones.
- SQLite3, base de données dédiée aux environnements embarqués est aussi présente pour fournir une capacité de stockage aux applicatifs.
- Google Maps, accès au service de Google intégrable directement dans les applications.

La présence des ces librairies rassure quant à la capacité de portage sur Android et offre un contexte très riche pour le développement d'application. La plate-forme propose un troisième groupe d'interfaces Java qui est le plus attendu par les développeurs, celui qui gère les applications sous Android ainsi que les relations homme/machine.

Des applications égales entre elles

La règle qui veut que tout développement sur Android se fasse en Java s'applique à toutes les applications et donc également aux applications que l'on considère habituellement comme "réservées", comme la ges-



tion de la téléphonie ou le browser Web. Google a écrit comme principe : sous Android, toutes les applications sont égales.

Une application Android se présente sous la forme d'un seul fichier d'extension ".apk" qui contient l'ensemble des ressources nécessaires à son fonctionnement.

L'application est décrite dans *AndroidManifest.xml* qui est contenu dans le fichier ".apk". Ce manifeste contient la définition des droits liés à l'application et énumère les différents composants de l'application ainsi que leurs règles d'activation.

Une application combine quatre différentes façons d'implémenter des traitements, qui correspondent chacune à une classe de base de la plateforme :

- **Activity** : l'activité correspond à une logique d'interaction entre l'utilisateur et le téléphone, en pratique c'est un écran.
- **Service** : le service permet d'implémenter des fonctionnalités s'exécutant en arrière plan sur une longue durée de temps.
- **IntentReceiver** : le récepteur de demandes de traitement permet d'effectuer des traitements sans que l'application soit nécessairement active.
- **ContentProvider** : le fournisseur de contenu permet de fournir l'accès à une ressource qui est partagée entre plusieurs applications. La ressource peut être une base SQLite, un ensemble de fichiers, des données sur le réseau ...

Le déclenchement des *Activity*, *Service* et *IntentReceiver* passe par l'utilisation d'*Intent* qui sont des demandes de traitement. Les demandes de traitement contiennent : une action, une catégorie d'action et éventuellement des données complémentaires. Les *Intent* peuvent être applicatifs ou systèmes, ils peuvent être à destination d'une classe spécifique ou être broadcastés. Voici l'exemple d'un *IntentReceiver* qui reçoit un *Intent* de notification de SMS et démarre l'*Activity* de lecture de ce SMS par l'envoi d'un nouvel *Intent* :

```
public class SMSIntentReceiver extends IntentReceiver {

    public void onReceiveIntent(Context context, Intent intent) {
        if (intent.getAction().equals(context.getString(R.string.sms_action))) {
            Bundle bundle = intent.getExtras(); // récupère le SMS
            abortBroadcast(); // arrête la diffusion de l'intent
            Intent intentN = new Intent(context, SMS.class);

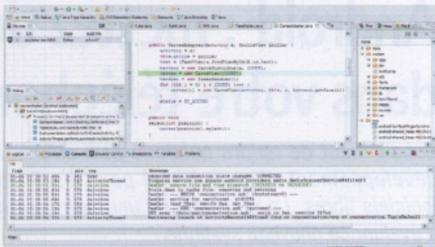
            // communique le SMS à l'activité
            intentN.setAction(context.getString(R.string.sms_action));
            intentN.putExtras(bundle); // bundle contient
            les données du SMS
            intentN.setLaunchFlags(Intent.NEW_TASK_LAUNCH);

            // assure le démarrage
            context.startActivity(intentN);

            // démarre l'activité
        }
    }
}
```

La réception de l'*Intent* a lieu dans l'*Activity* à son démarrage, le message SMS est récupéré à partir des données complémentaires de l'*Intent* :

```
public void onCreate(Bundle icicle) {
    super.onCreate(icicle);
    setContentView(R.layout.main);
}
```



```
Intent intent = this.getIntent();
if ((intent.getAction() != null) && intent.getAction().equals(this.get
String(R.string.sms_action) && (intent.getExtras() != null)) {
    SmsMessage[] messages =
        Telephony.Sms.Intents.getMessagesFromIntent(intent);
    TextView text = (TextView)this.findViewById(R.id.text);

    for (int i = 0; i < messages.length; i++){
        text.append("Message reçu de : <");
        text.append(messages[i].getDisplayOriginatingAddress());
        text.append(">\n");
        text.append(messages[i].getDisplayMessageBody());
        text.append("\n");
    }
} else {
    finish(); // pas besoin d'ouvrir l'activité quand pas de message
}
}
```

Le choix de la classe qui va recevoir le SMS est effectué par le système à partir des informations présentes dans *AndroidManifest*. Celui-ci contient un *IntentFilter* qui permet d'indiquer qu'une classe est prête à recevoir un *Intent* particulier :

```
<receiver class="SMSIntentReceiver">
    <intent-filter>
        <action android:value="android.provider.Telephony.SMS_RECEIVED" />
    </intent-filter>
</receiver>
```

Comme le *SMS_RECEIVED* est soumis à des droits, l'*AndroidManifest* doit déclarer son usage dans le manifeste :

```
<uses-permission id="android.permission.RECEIVE_SMS" />
```

Le mécanisme de démarrage de traitements par *Intent* est une généralisation des mécanismes de démarrage. Il apporte un réel confort à l'utilisation et standardise le démarrage, que ce soit à l'initiative d'une application ou à l'initiative d'un événement système, renforçant l'égalité entre les applications.

Les relations entre applications sont d'autant plus formalisées que chaque application tourne sur une machine virtuelle qui s'exécute dans un processus indépendant. Cela oblige pour la communication entre applications à utiliser un mécanisme de communication entre proces-



Abonnez-vous :
Le magazine mensuel
11 numéros par an : **45 €***
soit **3 Numéros GRATUITS**

*Tarifs France métropolitaine

Programmez ! est le magazine du développement
Langage et code, développement web, carrières et métier : *Programmez, c'est votre outil de veille technologique.*
Pour votre développement personnel et professionnel, abonnez-vous à Programmez !

PROGRAMMEZ

- Abonnement 1 an au magazine : 45 € (au lieu de 65,45 € tarif au numéro) Tarif France métropolitaine
 - 1 an au magazine + 5 numéros HS .NET : 63 € Tarif France métropolitaine
 - Abonnement Intégral : 1 an au magazine + archives sur Internet et PDF : 57 € Tarif France métropolitaine
 - Abonnement PDF / 1 an : 30 € - Tarif unique Inscription et paiement exclusivement en ligne www.programmez.com
 - Abonnement Etudiant : 1 an au magazine : 39 € (au lieu de 65,45 € tarif au numéro) Offre France métropolitaine
- PROGRAMMEZ HORS SERIE .NET** Seulement
- Abonnement 1 an aux Hors Série : 5 numéros : 20 € (au lieu de 25 € tarif au numéro) Tarif France métropolitaine



Abonnez-vous :
Les Hors-Série .Net
5 numéros par an : **20 €***
soit **1 Numéro GRATUIT**

*Tarifs France métropolitaine

+ Abonnement INTÉGRAL NOUVEAU

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

Cette option est réservée aux abonnés pour 1 an au magazine, quel que soit le type d'abonnement (Éco, Numérique, Etudiant). Le prix de leur abonnement normal est majoré de 12 € (prix de

lancement, identique pour toutes zones géographiques). Pendant la durée de leur abonnement, ils ont ainsi accès, en supplément, à tous les anciens numéros et articles /dossiers parus.

OUI, je m'abonne

Vous pouvez aussi vous abonner en ligne et trouver tous les tarifs www.programmez.com

PROGRAMMEZ

- Abonnement 1 an au magazine : 45 € (au lieu de 65,45 € tarif au numéro) Tarif France métropolitaine
- 1 an au magazine + 5 numéros HS .NET : 63 € Tarif France métropolitaine
- Abonnement Intégral : 1 an au magazine + archives sur Internet et PDF : 57 € Tarif France métropolitaine
- Abonnement Etudiant : 1 an au magazine : 39 € (au lieu de 65,45 € tarif au numéro) Offre France métropolitaine

PROGRAMMEZ HORS SERIE .NET NOUVELLE OFFRE

- Abonnement 1 an aux Hors Série : 5 numéros : 20 € (au lieu de 25 € tarif au numéro) Tarif 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

Programmez!
LE MAGAZINE DU DÉVELOPPEMENT

Offre limitée,
valable jusqu'au
30 avril 2008

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

Conformément à la loi Informatique et Libertés du 06/07/78, vous disposez d'un droit d'accès et de rectification aux données vous concernant.

Par notre intermédiaire, nous pourrions être amenés 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.

sus. Pour cela, Android fournit un mécanisme de communication, similaire à Corba, avec un formalisme de description des interfaces: AIDL (Android Interface Definition Language) et une sérialisation qui est d'ailleurs différente de la sérialisation Java habituelle. Ce choix d'architecture garantit que les applications échangeront des données entre elles, comme elles le feront avec des processus systèmes écrits en C. Du fait du confinement d'une application Java dans un processus Unix, Dalvik peut offrir aux applications un modèle de sécurité basé sur celui de Linux.

L'adoption d'un même modèle de sécurité entre Java et Linux permet de réaliser en Java des applications systèmes qui pourront avoir des relations contrôlées avec des objets systèmes protégés. Une autre des possibilités novatrices d'Android est la gestion des interactions avec l'utilisateur. Nous allons illustrer quelques possibilités de la plate-forme par la réalisation d'un jeu de concentration consistant à découvrir des paires de cartes identiques.

Des ressources pour concevoir les écrans

Pour un développeur Java, habitué à Swing, AWT ou MIDP, qui découvre la façon de concevoir des IHM avec Android, une des nouveautés la plus intéressante est la généralisation de fichiers ressources.

Les ressources Android gèrent les objets de base comme les *String*, *Color*, *Dimension* et *Drawable* qui représentent les ressources graphiques. Les *Drawable* sont soit des images, soit des composés plus évolués comme les *shape* qui permettent d'arrondir les coins et de définir le motif de l'arrière des cartes :

```
<shape xmlns:android="http://schemas.android.com/apk/res/android"
type="oval">
<solid android:color="#ff000000"/>
<padding android:left="2dp" android:top="2dp"
android:right="2dp" android:bottom="2dp" />
<corners android:radius="4dp" />
</shape>
```

Les ressources peuvent être référencées, soit à l'intérieur des fichiers ressources :

```
android:background="@drawable/bkgd_global"
```

soit dans Java, le lien avec le code étant fait par l'intermédiaire d'une classe R (comme Ressource) qui contient les références aux différentes ressources :

```
public CarteView(Context context, Verificateur verificateur, int num,
Drawable carte) {
super(context);
...
setBackground(context.getResources().getDrawable(R.drawable.bkgd_carte));
...
}
```

Les ressources peuvent gérer des éléments évolués comme l'animation. Par exemple, la découverte d'une carte est la combinaison d'une animation de disparition :

```
<scale xmlns:android="http://schemas.android.com/apk/res/android"
android:fromXScale="1.0" android:toXScale="1.0"
```

```
android:fromYScale="1.0" android:toYScale="0.0" android:duration
="500"
android:pivotX="50%" android:pivotY="50%" />
```

et d'apparition.

```
<scale xmlns:android="http://schemas.android.com/apk/res/android"
android:fromXScale="1.0" android:toXScale="1.0"
android:fromYScale="0.0" android:toYScale="1.0" android:duration
="500"
android:pivotX="50%" android:pivotY="50%" />
```

Le déclenchement de l'animation décrite dans les ressources est effectué par la classe *FlipperView* :

```
setInAnimation(AnimationUtils.loadAnimation(context, R.anim.carte_in));
setOutAnimation(AnimationUtils.loadAnimation(context, R.anim.carte_
out));
```

Les ressources permettent aussi de manipuler les composants graphiques de haut niveau qui sont basés sur la classe *View* d'Android (cette classe est l'équivalent de *Component* sous Swing ou de *Container* sous AWT). L'utilisation de ressources permet de réduire au minimum le code des activités. La disposition des cartes sur le tapis, est gérée par une activité :

```
package org.ot.concentration;
```

```
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemClickListener;
```

```
public class TapisDeJeu extends Activity {
CartesAdapter adapter;
```

```
@Override
protected void onCreate(Bundle icle) {
super.onCreate(icle);
```

```
setContentView(R.layout.concentration);
GridView grille = (GridView) findViewById(R.id.maGrille);
grille.setAdapter(adapter = new CartesAdapter(this, grille);
grille.setOnItemClickListener(new OnItemClickListener() {
public void onItemClick(AdapterView parent, View v, int position,
long id) {
adapter.select(position);
}
});
}
```

C'est *concentration.xml* (voir listing 1 sur le CD Rom/site web) qui décrit la vue et c'est l'adaptateur qui gère le lien entre les données et l'affichage. Le chargement des descriptions d'écrans depuis les ressources se fait par *setContentView()*. L'interaction entre le code et les composants

décrits dans les ressources s'effectue en récupérant les composants par `findViewById()`.

Le seul point faible de cette approche est qu'elle lie les descriptions d'écrans à l'Activity. Or certains composants peuvent utiliser des descriptions d'écrans en dehors des Activity, comme les `PopupWindow`. Les ressources Android peuvent être étendues. Par exemple, si l'on souhaite supprimer de la classe `GridView` les événements indésirables lors des mouvements sur le fond d'écran, on spécialise la classe de la façon suivante :

```
public class GrilleView extends GridView {
    public
    GrilleView(Context context, AttributeSet attrs, Map params) {
        super(context, attrs, params);
    }
    @Override
    public boolean onMotionEvent(MotionEvent event) {
        return true;
    }
}
```

Cette classe peut être utilisée directement dans le fichier de ressources en lieu et place de la `GridView` :

```
<org.ot.memoire.GrilleView
    id="@+id/moGrille"
    android:layout_width="fill_parent"
```

```
android:layout_height="wrap_content"
android:layout_weight="1"
android:columnWidth="40"
android:gravity="center"
android:focusable="false"
android:background="@drawable/bkgd_global"
/>
```

L'utilisation des ressources apporte aux composants graphiques très riches d'Android une réelle facilité de mise en oeuvre.

Une plate-forme pour les développeurs

Si le SDK Android possède encore quelques défauts de jeunesse, principalement au niveau de la documentation, les développeurs ne peuvent qu'être séduits par les fonctionnalités de la plate-forme, le niveau de sophistication des interfaces graphiques, les principes guidant la gestion des applications et le parti pris de l'Open Source.

Tous ces ingrédients ajoutés à la puissance marketing de Google feront à coup sûr de cette nouvelle plate-forme un acteur incontournable du logiciel sur les mobiles. Il lui reste encore plusieurs défis à relever: celui, le plus important, de la première implémentation dans un terminal, et celui de l'adoption par les constructeurs. Mais nul doute que les capacités d'Android permettront de construire des Smartphones à l'égal de l'iPhone, des Blackberry ou du N95.

■ Olivier Théry

Un environnement de développement mature

L'existence d'Android n'a pas commencé par une implémentation du système dans un modèle de Smartphone ou de PDA mais par la mise à disposition d'un environnement de développement complet. Ce SDK est disponible en téléchargement sur <http://code.google.com/android>.

Le développement sous Android s'effectue de la même façon que pour le développement Java, le SDK utilise d'ailleurs le compilateur Java 5. Le SDK apporte des outils de mise au point comme le simulateur de mobile (**emulator**). Ce composant, basé sur l'émulateur **qemu**, permet d'offrir sur un PC l'environnement logiciel d'Android en émulant le processeur ARM. L'utilisation d'un émulateur de processeur permet d'exécuter le code Android authentique et ainsi d'obtenir un excellent niveau de compatibilité entre la plate-forme où a lieu la simulation et la plate-forme cible. L'émulateur est couplé à un autre outil : **adb** (Android Debug Bridge). **Adb** fait le lien entre la plate-forme de développement et la plate-forme cible. Il permet, entre

autres, l'échange de fichier, le démarrage d'un Shell sous Linux,...

D'autres outils existent pour :

- Construire les fichiers applications (*apk*) à partir des fichiers *.class* produits par le compilateur Java.
- Accéder aux fichiers de données *sqlite3*
- Construire les systèmes de fichiers "yaffs2"
- ...

Mais l'outil le plus utile est sans nul doute le plug-in Eclipse (pas de plug-in officiel pour d'autres IDE, ndlr) qui permet de développer des applications Android de la même façon que des applets ou des servlets. Le développeur peut bénéficier de toutes les fonctionnalités de l'IDE que ce soit pour l'édition, la gestion des sources, les tests ou la mise au point. L'interface est enrichie avec des fonctionnalités liées aux particularités de la plate-forme :

- Un contrôleur permettant de simuler des appels téléphoniques et d'envoyer des SMS.
- Des fenêtres permettant de visualiser l'état des processus, leur heap et threads.



- Une fenêtre de contrôle du simulateur permettant de prendre des photos d'écran.
- Une fenêtre sur la trace système et applicative, permettant de récupérer la sortie du *Log.v()* qui remplace le *System.out*.
- Un navigateur dans le système de fichier du simulateur.

En résumé, le SDK fournit un environnement de développement très complet qui s'intègre "sans couture" avec Eclipse. Il permet au développeur Java d'être immédiatement opérationnel sur Android (rappelez-vous qu'il s'agit d'un pré-version et des problèmes de stabilité peuvent apparaître, ndlr).

Les Shaders et la 3D temps réel

La technologie des Shaders prend aujourd'hui une place très importante dans le domaine de la programmation 3D temps réel. Je vous propose dans cet article de vous présenter le fonctionnement de ces programmes directement exécutés par la carte graphique.

Les shaders sont utilisés dans de nombreux effets spéciaux en 3D temps réel, par exemple les calculs d'éclairages évolués (l'ombrage des objets ainsi que la diffusion, la réflexion et la réfraction de la lumière), ou la réalisation de brouillard, d'explosions. Un shader (du verbe shade : ombrager, estamper, nuancer) est un programme utilisé dans la 3D temps réel pour paramétrer une partie du processus de rendu. Cette technologie restait - jusqu'à l'arrivée des langages dernières générations - réservée à quelques spécialistes. Aujourd'hui, avec des langages, dont la syntaxe est proche du C, et des outils adaptés, ces programmes peuvent être écrits par tout développeur possédant quelques notions de base en 3D.

Historique

Introduit sur le marché par Pixar lors de la sortie de son logiciel RenderMan, les shaders se sont répandus au fur et à mesure que le prix des composants graphiques diminuait. Initialement destinés à l'image de synthèse, les shaders sont introduits dans les technologies de programmation 3D temps réel en 2001 (date de sortie de DirectX 8 et OpenGL 1.4). Ils sont depuis supportés par les grands constructeurs de cartes graphiques. Deux langages ont réussi à s'imposer pour l'écriture de shaders 3D temps-réel :

• Cg Shader Language

Développé par Nvidia, ce langage (livré avec un certain nombre d'outils gratuits), a été conçu pour permettre la production intégrée. Il a bien survécu à l'apparition des langages de shaders dernière génération en s'imposant notamment sur la plate-forme Playstation 3.

• DirectX High-Level Shader Language (HLSL)

C'est le langage le plus populaire, principalement de par le fait qu'il se rapproche du langage C (A contrario des langages de type assembleur comme GLSL et Cg). Avant de voir quelques exemples de shaders écrits en HLSL, il convient de mettre au point quelques notions sur le fonctionnement des processeurs de cartes graphiques. Un exemple de shaders Fig.1.

Les processeurs de cartes graphiques

On désigne par GPU le processeur de la carte graphique (Graphic Processing Unit = unité de traitement graphique). Son rôle est de produire, pour un ensemble de points représentant un monde en 3D, la couleur de chaque pixel de la zone de rendu. Les GPU ont été spécialement conçus et optimisés pour effectuer efficacement les calculs relatifs à la 3D (principalement des calculs matriciels). Ils diffèrent donc des CPU qui peuvent effectuer tous types d'opérations mais qui sont beaucoup moins rapides que les GPU dans le domaine des calculs géométriques. Le GPU est constitué de nombreux composants, les deux plus importants étant le vertex pipeline et le pixel pipeline car ils sont programmables (les autres sont paramétrables uniquement). Le premier s'occupe des vertices (pluriel de vertex) et le second des pixels. Dans un premier temps nous pouvons faire l'abstraction suivante pour bien

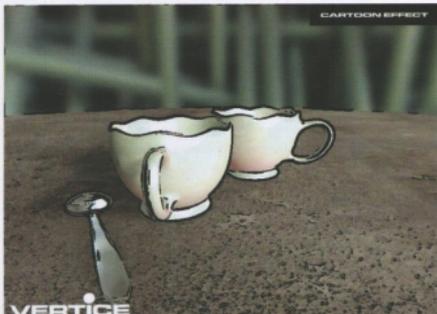


Fig.1

comprendre la différence entre ces 2 pipelines : le vertex pipeline effectue les traitements relatifs à la géométrie alors que le pixel pipeline effectue les traitements relatifs au rendu de la scène (les pixels à afficher à l'écran).

Le pipeline (c'est-à-dire l'ensemble des traitements ordonnés effectués d'un GPU) est complexe. On peut cependant le schématiser pour en comprendre le fonctionnement :

1. Les données géométriques sont passées en entrée du pipeline (une scène 3D sous forme de points constitués de 3 coordonnées, plus éventuellement des coordonnées de texture et de normales, etc.).
2. Le **vertex pipeline** qui effectue les transformations géométriques souhaitées (transformation des objets du repère monde dans le repère écran au minimum le plus souvent)
3. Une série d'opérations va permettre ensuite d'éliminer des points pour ne garder que ceux qui seront "transformés en pixel" (c'est-à-dire ceux qui seront traités par le **pixel pipeline**). Voici, ci-dessous, un exemple des opérations réalisées dans cette phase :
 - **Clipping** : Suppression des points qui ne sont pas dans le champ de vision de la caméra (il est inutile de calculer la couleur des points qui ne seront pas visibles par la caméra)
 - **Culling** : Elimination des faces arrières. On ne va pas calculer la couleur de l'arrière des faces car elles ne sont pas visibles du point de vue de l'utilisateur (sauf paramétrage)
 - **Tramage** : Construction de primitives à partir des vertices (on crée des primitives plus "lisses" en interpolant certaines informations comme la couleur). Le tramage crée un certain nombre de pixels qui vont être envoyés au pixel pipeline pour traitement.
4. Le **pixel pipeline** reçoit en entrée les données issues du tramage et détermine pour chaque pixel, la couleur à afficher.

5. Avant d'effectuer le rendu proprement dit, le GPU effectue les tests d'alpha et de profondeur, puis effectue les opérations d'alpha blending (gestion de la transparence) et de brouillard.

Construire un shader, c'est spécifier au GPU les opérations à effectuer au niveau du vertex pipeline et du pixel pipeline (on a, de plus, la possibilité de configurer le *culling*, l'*alpha blending* et de paramétrer les tests d'alpha et de profondeur). Le code d'un shader est directement interprété par le GPU.

Avec les vertex shaders, on déconnecte le vertex pipeline par défaut pour imposer le nôtre. De même, avec les pixels shaders, on déconnecte le pixel pipeline par défaut. On a la possibilité d'effectuer ce traitement par objet, c'est pour cela que l'on peut considérer les shaders comme des matériaux.

Des outils pour tester vos shaders

Lorsque l'on développe des Shaders, il est nécessaire de s'équiper afin de pouvoir tester leur rendu et leur bon fonctionnement. Vous trouverez dans le tableau ci-dessous une liste d'outils disponibles.

Nom	Description	Lien
Shader Forge NOVA 2008	La Shader forge est un des outils disponibles dans la version 2008 du moteur 3D temps réel NOVA (réalisé par la société toulousaine Vertice). Cet outil possède l'avantage de pouvoir interpréter automatiquement les instructions SAS (la norme SAS introduite par Microsoft permet la liaison entre la scène et les paramètres des shaders) et permet de tester en temps réel, durant la phase de développement, le rendu des shaders.	http://www.vertice.fr
FX Composer	FX Composer possède l'avantage de pouvoir créer des shaders HLSL et Cg sous DirectX et OpenGL.	http://developer.nvidia.com/object/fx_composer_home.html

Programmation des GPU

Vertex shader et Pixel shader de base

L'approche par fichier d'effets permet de spécifier dans le même fichier les programmes des vertex shaders et des pixels shaders. On regroupe ensuite les vertex et pixel shaders en passes et en *technics* :

Une *pass* représente un rendu de la scène par le shader, elle spécifie les points d'entrée des vertex et pixel shaders. Il est important de noter qu'on peut avoir besoin de plusieurs rendus d'une même scène pour obtenir certains types d'effets.

Une *technic* est composée d'une ou plusieurs passes et permet de la configuration du pipeline de rendu graphique (par exemple les valeurs à utiliser pour le test alpha, le test de profondeur, etc.).

Structure de base d'un programme écrit en HLSL :

```
// Rappel : un vertex = un sommet d'une face.
// Représente la matrice de transformation dans le repère de l'écran
float4x4 WorldViewProjection;
// Structure des vertices en entrée du vertex shader
struct VS_INPUT
{
    // Au minimum on devra transformer la position du vertex dans le repère de l'écran.
```

```
float4 position;
};
// Structure des vertices en sortie du vertex shader (et donc en entrée du
pixel shader)
struct VS_OUTPUT
{
    // La position du vertex dans le repère du monde
    float4 position;
};
// Le vertex shader
VS_OUTPUT main(VS_INPUT In)
{
    // On déclare ce que l'on va retourner
    VS_OUTPUT Out;
    // Transformation du vertex dans le repère de l'écran
    Out.position = mul(In.position, WorldViewProjection);
    // On retourne notre vertex, Out est passé au pixel shader
    return Out;
}
// Entre le vertex shader et le pixel shader, il y'a une interpolation des
vertex pour remplir les faces : on passe de vertex à pixel
// Le pixel shader
float4 mainPS(VS_OUTPUT In)
{
    // On retourne blanc comme couleur de ce pixel
    return 1.0f;
}
// Au moins une technique (technique = assemblage d'un vertex shader
et d'un pixel shader + configuration du pipeline)
// Au moins une passe par technique (passe = un rendu de la scène)
technique MainTechnique
{
    pass P0
    {
        vertexShader = compile vs_1_1 mainVS[];
        pixelShader = compile ps_1_1 mainPS[];
    };
}
```

Gérer l'éclairage d'un objet

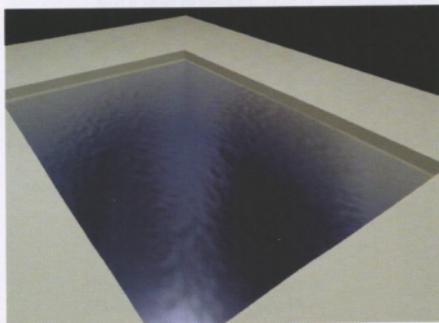
Comprendre comment fonctionne l'éclairage en 3D permet de bien comprendre ce qui se passe lorsqu'on programme des shaders.

Le calcul de l'éclairage peut s'effectuer de 2 manières différentes :

- Au niveau des sommets des faces (c'est à dire dans le vertex shader) ;
 - Au niveau des pixels directement (c'est à dire dans le pixel shader).
- La première méthode est plus rapide car le GPU effectue une interpolation du calcul d'éclairage pour les pixels de la face. Le rendu est cependant moins réaliste. Avec la deuxième méthode, le calcul de l'éclairage est effectué pour tous les pixels. Cela a un coût au niveau du GPU mais l'effet obtenu est plus réaliste.

En fonction des besoins et de l'effet souhaité, on adoptera l'une ou des 2 techniques. Les jeux de dernières générations utilisent au maximum les pixels shaders pour leurs effets spéciaux car ils permettent d'effectuer des calculs sophistiqués afin de déterminer la valeur de chaque pixel rendu à l'écran.

Un exemple de modification visuelle par les shaders :



Modèle d'éclairage

De nombreux modèles ont été élaborés. Un des plus couramment utilisé dans la programmation de shader est le suivant :

L'intensité lumineuse en un point est le produit scalaire de la normale en ce point par le vecteur direction de la lumière.

Une fois l'intensité lumineuse calculée en un point, il suffit de multiplier la couleur de ce point par l'intensité lumineuse pour appliquer l'effet d'éclairage. Nous allons, ci-dessous, mettre en place ce modèle d'éclairage à l'aide des shaders. Reprenons notre exemple précédent et modifions-le. La position de la lumière est une variable de notre shader :

```
float4 lightPosition ;
```

Nous souhaitons appliquer notre modèle d'éclairage sur un objet texturé (recouvert), on doit donc déclarer un sampler pour notre texture. La texture à utiliser dans notre shader :

```
texture MaTexture;  
Le sampler permettant d'y accéder :  
sampler Sampler1 = sampler_state  
{  
    texture = {MaTexture};  
};
```

Calcul de l'éclairage au niveau du Vertex Shader

Structure des points en entrée du vertex shader. Nous avons besoin de la normale au point :

```
struct VS_INPUT  
{  
    // Position du sommet  
    float4 position;  
    // Normale au point  
    float3 norm;  
    // Coordonnées de texture associées à ce sommet  
    float2 tex;  
};
```

En sortie du vertex shader (et donc en entrée du pixel shader), nous aurons la valeur de l'intensité de la lumière en ce point :

```
struct VS_OUTPUT  
{  
    float4 position;  
    float2 tex;  
    float lightIntensity;  
};
```

Dans le vertex shader :

```
VS_OUTPUT main(VS_INPUT In)  
{  
    VS_OUTPUT Out;  
    // Transformation du vertex dans le repère de l'écran  
    Out.position = mul(In.position, WorldViewProjection);
```

```
// Coordonnées de texture transmises au pixel shader
Out.tex0 = In.tex0;
// La normale du point est transformée dans le repère de l'écran (elle
"arrive" dans le repère local de l'objet) afin de pouvoir l'utiliser avec light
Direction : le vecteur de direction de la lumière (qui est exprimé lui, dans
le repère de l'écran)
float3 N = mul(In.norm, (float3x3)World);
// Vecteur direction de la lumière
float3 lightVector = P - lightPosition;
// On normalise les 2 vecteurs obtenus
N = normalize(N);
float3 lightDirection = normalize(lightVector);
// On effectue le produit scalaire grâce à l'instruction HLSL dot. L'intensité de
la lumière a été calculée pour ce sommet, elle est transmise au pixel shader.
Out.lightIntensity = dot(N, lightDirection);
return Out;
}

// Le pixel shader effectue la multiplication d'une couleur lue dans une
texture par l'intensité de la lumière.
float4 mainPS(VS_OUTPUT In)
{
    float4 color = tex2D(Sampler1, In.tex0);
    return color * In.lightIntensity;
}
```

Calcul de l'éclairage au niveau du Pixel Shader

La structure des points en entrée du vertex shader et le pixel shader.

```
struct VS_INPUT
{
    float4 position;
    float3 norm;
    float2 tex0;
};

struct VS_OUTPUT
{
    float4 position;
    float2 tex0;
    float3 norm;
    float3 light;
};
```

Le vertex shader :

```
VS_OUTPUT mainVS(VS_INPUT In)
{
    VS_OUTPUT Out;
    // Transformation du vertex dans le repère de l'écran
    Out.position = mul(In.position, WorldViewProjection);
    // Position dans le repère du monde
    float3 P = mul(In.position, World);
    // Normale dans le repère du monde...
    float3 N = mul(In.norm, (float3x3)World);
    // ...transmise au pixel shader :
    Out.norm = normalize(N);
```

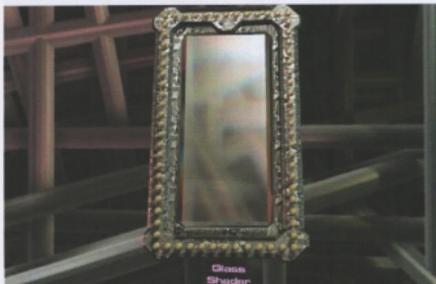
```
// Vecteur de direction de la lumière...
float3 lightVector = P - lightPosition
// ...normalise et transmise au pixel shader
Out.light = normalize(lightVector);
return Out;
}
```

Le calcul du produit scalaire est effectué dans le pixel shader. L'intensité n'est pas calculée au niveau des sommets mais bien pour tous les pixels. Il n'y aura pas d'interpolation comme avec le vertex shader. L'effet est beaucoup plus précis :

```
float4 mainPS(VS_OUTPUT In)
{
    float intensity = dot(In.norm, In.light);
    float4 color = tex2D(Sampler1, In.tex0);
    return color * intensity;
}
```

Le comportement de notre éclairage peut maintenant facilement être modifié. Si on veut que les points directement en face de la lumière soient tous blancs, alors on fera par exemple :

```
float4 mainPS(VS_OUTPUT In)
{
    float intensity = dot(In.norm, In.light);
    float4 color = tex2D(Sampler1, In.tex0);
    // ...Si l'intensité est supérieure à 0,5 : affiche du blanc. Sinon
: affiche la texture influencée par la lumière.
return (intensity > 0.5)? 1.0f : color * intensity;
}
```

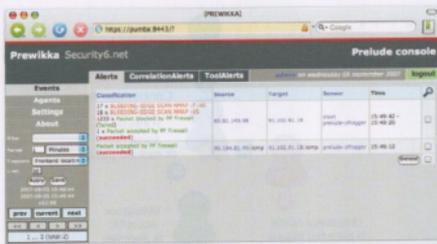


Un exemple de shaders prenant en compte les notions ci-dessus :

Conclusion

Les shaders sont aujourd'hui une brique incontournable du développement 3D. A l'origine de beaucoup d'effets visuels, ils sont faciles à aborder et de nombreux outils accompagnent leur développement.

■ **Fabien Le vavasseur**
 Vertice - Fabien.levavasseur@vertice.fr



Prelude IDS

Le format IDMEF

IDMEF signifie "Intrusion Detection Message Exchange Format", qui se traduit en "Format d'Échange de Messages de Détection d'Intrusions". Il s'agit d'une norme disponible au sein de l'IETF sous forme de RFC. Le format de messages est XML, qui offre de nombreux avantages :

- Normalise les événements
- Étant un format XML, il permet de valider le message avec une DTD
- Unifie l'interaction entre les différents IDS sans se soucier de leur nature
- Fournit un vocabulaire précis pour les développeurs et utilisateurs de systèmes de détection d'intrusions

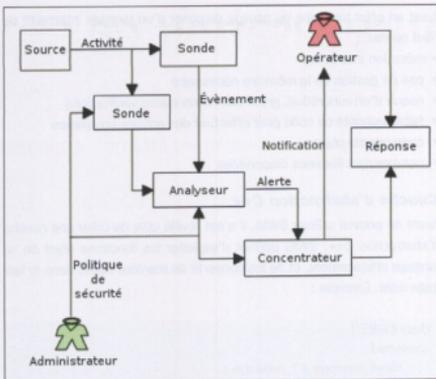
Le format XML n'est cependant pas toujours adapté (notamment pour les performances), c'est pourquoi les messages sont souvent stockés dans un format binaire, et IDMEF sert principalement à l'échange des messages de manière standard. Son modèle de données permet de faire face aux problématiques suivantes :

- **Les informations sur une alerte sont hétérogènes** : le format IDMEF est orienté objet selon la masse d'information que la sonde peut collecter
- **Les environnements d'opération des IDS sont multiples** : la détection d'intrusions réseau diffère de la détection d'intrusions machine, à commencer par la fiabilité des événements qu'elle peut collecter
- **Les capacités des analyseurs sont différentes** : en fonction de votre réseau, vous pouvez très bien limiter votre IDS pour ne détecter qu'un certain type d'attaques. Étant maître de votre réseau, il s'avère important de chercher à enrichir le message d'origine
- **Les environnements d'opération sont différents** : Unix, Linux et Syslog ne fournissent pas le même type de log qu'une machine Windows
- **Les objectifs commerciaux sont différents** : ce sont parfois les objectifs commerciaux qui dirigent les objectifs techniques, le format permet d'enrichir ces informations pour les exploiter par la suite

Dans IDMEF, chaque valeur est associée à un chemin, c'est-à-dire la séquence de clés permettant d'arriver jusqu'à l'élément. Certains éléments peuvent être des listes (la destination de l'attaque, par exemple). Ce chemin permet de décrire facilement et de manière unique chacun des critères d'une attaque : le message de description se retrouve dans le chemin `alert.classification.text`, la sévérité potentielle dans `alert.assessment.impact.severity`, etc. Ce sont ces critères que nous allons représenter graphiquement.

La normalisation

L'utilisation d'un vocabulaire précis est importante : il permet aux utilisateurs de bien comprendre et aux développeurs de bien structurer leur programme. Passer le vocabulaire en revue nous permet aussi de com-



prendre les différentes fonctionnalités d'un système de détection d'intrusions réseau complet.

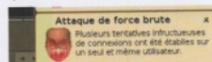
Nous vous proposons de suivre le chemin de vie d'une alerte, de sa naissance jusqu'à son enterrement.

Une alerte doit être conçue par trois éléments. Le premier est une **source** de données, qui est en fait une activité pouvant être une ligne de log, une description de politique de sécurité ou encore un paquet réseau. Une fois cette source reçue par la **sonde**, elle est envoyée à l'**analyseur** en vue d'extraire ce qui nous intéresse depuis la **source**, selon en général une suite de signatures.

Ensuite, les **analyseurs** envoient leurs résultats sous forme d'alerte IDMEF au **concentrateur**. Ce **concentrateur** stocke alors ces alertes en général dans une base de données. Il peut notifier l'**opérateur**, qui est en fait l'analyste de sécurité. Cet analyste peut donner des indications permettant d'avoir une **réponse** fiable face à la menace détectée.

La notification est intéressante, elle peut être filtrée par la corrélation (fait de regrouper un ensemble d'événements pour en déduire des alertes plus ciblées). Ainsi, comme dans l'image précédente, on peut remarquer une attaque correspondant non pas au fait qu'une connexion a échoué sur un utilisateur, mais sur une succession d'échecs. Ce qui correspond à un pirate cherchant certainement à essayer toute une série de mots de passe.

Nous allons ainsi récupérer la source, le type d'attaque et la destination. Mais avant d'arriver jusque là, nous allons vous présenter SWIG, qui nous a permis de modifier la libprelude afin de pouvoir utiliser facile-



ment Python et créer un programme qui fait ce que nous voulons en quelques lignes de code.

SWIG

SWIG (Simplified Wrapper and Interface Generator) est un outil permettant d'encapsuler un langage dans un autre, pour permettre par exemple d'utiliser des fonctions définies dans un langage bas niveau tel que C, dans un langage de haut niveau, tel que Python ou Perl. SWIG supporte la conversion vers une dizaine de langages environ.

Nous allons utiliser SWIG pour créer une couche d'abstraction autour de la librairie Prelude. Pour pouvoir effectuer des développements rapides,

il est en effet très utile de pouvoir disposer d'un langage interprété de haut niveau :

- exécution interactive
- pas de gestion de la mémoire nécessaire
- risque d'erreurs réduit, grâce aux nombreuses vérifications
- faible quantité de code pour effectuer des actions complexes
- code orienté objet
- nombreuses bibliothèques disponibles

Couche d'abstraction C++

Avant de pouvoir utiliser SWIG, il s'est révélé utile de créer une couche d'abstraction C++. SWIG permet d'exploiter les fonctions objet de ce langage efficacement, et de les convertir de manière native dans le langage cible. Exemple :

```
class IDMEF {
protected:
    idmef_message_t *_message;
public:
    IDMEF() throw (const char *);
    int Set(const char *object, const char *value);
    char * Get(const char *object) throw (const char *);
};
```

Chaque fonction effectue des opérations complexes (ici, création d'une valeur IDMEF associée à un chemin)

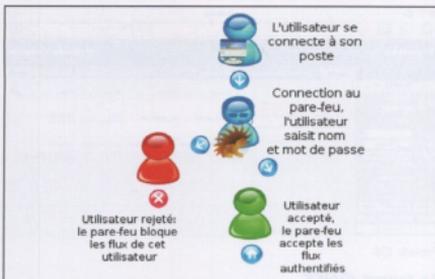
```
int IDMEF::Set(const char *object, const char *value)
{
    int ret;
    idmef_path_t *path;
    idmef_value_t *val;
    ret = idmef_path_new(&path, object);
    if (ret < 0)
        return ret;
    ret = idmef_value_new_from_path(&val, path, value);
    if (ret < 0) {
        idmef_path_destroy(path);
        return ret;
    }
    ret = idmef_path_set(path, _message, val);
    idmef_value_destroy(val);
    idmef_path_destroy(path);
    return 0;
}

utilisation-de-swig [1] Utilisation de SWIG utilisation-de-swig
```

Utilisation de SWIG

SWIG utilise un fichier décrivant les fonctions à encapsuler. De manière typique, ce fichier va inclure le code C++

```
%
module PreludeEasy
%{
#include "easy-idmef.h"
%}
#include easy-idmef.h
```



Nu FW (Windows)

Ce fichier permet de contrôler de manière précise le code que SWIG va générer, s'il y a des conversions de type de données à effectuer, etc. La compilation de ce fichier par SWIG donne plusieurs fichiers (ici, en Python) :

```
swig -c++ -DPRELUDE_
PYTHON -python -o easy-idmef_wrap_python.cxx swig/libprelude-easy.i
```

On obtient deux fichiers :

- `easy-idmef_wrap_python.cxx` : le code C++ du module, qu'il faut compiler
- `PreludeEasy.py` : le code python du module, qui fait le lien avec ce code C++

Utilisation du module

Après compilation des fichiers obtenus par SWIG, le module est prêt à être installé et utilisé. On l'utilise comme un module standard Python

```
import PreludeEasy
idmef = PreludeEasy.IDMEF()
idmef.Set("alert.classification.txt", "Message")
print idmef.Get("alert.classification.txt")
```

Le même code, en langage C, aurait pris plusieurs dizaines de lignes ! Grâce à l'utilisation de SWIG, nous pouvons désormais exploiter la puissance du langage Python pour développer rapidement un programme combinant Prelude et des bibliothèques de visualisation 3D. Attention tout de même, les bindings sous leur forme actuelle sont encore en phase de développement intensif et ne sont supportés que de manière expérimentale par le projet Prelude.

- **Pierre Chifflier** - Ingénieur-Chercheur - INL
- **Sébastien Tricaud** - Ingénieur-Chercheur - INL

- (1) <http://www.prelude-ids.org>
- (2) <http://tools.ietf.org/rfc/rfc4767.txt>
- (3) <http://www.swig.org>
- (4) <http://www.python.org>
- (5) <http://www.graphviz.org>
- (6) <http://www.secdev.org/projects/rtgraph3d>
- (7) <http://www.prelude-ids.com>
- (8) http://fr.wikipedia.org/wiki/Système_de_détection_d'intrusion
- (9) <http://www.wzdftpd.net/trac/wiki/Misc/PreludeEasy>

La programmation du port série du PC avec Java

Tels des sujets d'examens de fin d'année, certains thèmes de programmation reviennent périodiquement. Ainsi en est-il de la programmation du port série du PC. Nous abordons aujourd'hui cette question avec Java. Une bonne occasion de constater qu'écrire du code portable reste un vrai défi !)

La programmation du port série est un vaste sujet. En plus de ses difficultés propres, les particularités de tel ou tel périphérique, langage, ou plate-forme viennent ajouter leurs lots de chausse-trappes. Cet article va illustrer cela avec Java sous Linux et Windows Vista. Le lecteur trouvera les sources complètes des exemples sur le Cd-Rom accompagnant le magazine ou sur notre site officiel.

1 Matériel et logiciels

Voici avec quoi votre serveur a travaillé lors de la préparation de cet article :

	Windows	Linux
Système	Windows Vista	Gentoo, noyau 2.6.21-gentoo-r4
JDK	JDK 1.6.0_02-b06 Sun	JDK build 1.6.0-b105 Sun
API Java Communications	Javacomm 2.0/Win 32	Comm 3.0 u1 Linux
Périphérique	Modem U.S. Robotics 56K FAX EXT Rev. 11.6.11	Modem U.S. Robotics 56K FAX EXT Rev. 11.6.11
Outil de développement	Netbeans 5.5.1	Netbeans 5.5

J'ai tenu à détailler ce matériel très précisément afin que chacun puisse s'il lui souhaite vérifier par lui même certains comportements pour le moins surprenants, observés lors des essais et rapportés dans cet article. Faisons encore quelques commentaires. Sous Linux, le noyau doit être compilé avec le support du port série, comme ceci :

```
Device Drivers -->
Character devices -->
Serial drivers -->
[ * ] 8250/16550 and compatible serial support
[ 1 ] Console on 8250/16550 and compatible serial port
[ 4 ] Maximum number of 8250/16550 serial ports
[ 4 ] Number of 8250/16550 serial ports to register at runtime
```

Il est sage de demander, comme ci-dessus, que le support soit intégré au noyau plutôt que chargé depuis un module. Et bien sûr, la première chose à faire avant tout essai sous Linux est de vérifier que le noyau est bien configuré convenablement.

La JVM Java ne peut travailler seule avec le port série. Elle doit pour cela utiliser une librairie écrite en code natif interfacée avec des classes Java. C'est exactement ce que propose l'API Java Communications. Mais il y a déjà un petit souci. Au moment de la rédaction de cet article, le site de Sun Microsystems ne propose aucun téléchargement de cette API pour Windows. Cela ne serait plus possible, apprend-on, en raison de "certaines modifications" apportées par Microsoft au système Win-

downs. Donc pas d'API 3.0 pour Windows et même pas d'API 2.0. En cherchant sur Internet, j'ai fini par trouver un lien (compliqué) qui vous permettra de télécharger l'API 2.0. Vous trouverez ce lien dans l'archive de sources. En dépit des "modifications apportées à Windows" cette API fonctionne parfaitement sous Windows, y compris Vista. On ne peut pas en dire autant de l'API 3.0 pour Linux que vous téléchargerez normalement depuis le site de Sun Microsystems.

Le périphérique est un modem externe, j'ai choisi ce type de périphérique parce qu'il est courant et peu cher. En outre les commandes Hayes sont standard. Le lecteur qui étudiera les sources avec attention constatera que les commandes Hayes envoyées au modem sont terminées par le caractère `\r` (retour chariot) seulement et non pas par la séquence `\r\n`. En effet et contrairement à ce qu'on peut lire généralement, y compris sur Wikipédia, une commande Hayes est terminée par `\r` seulement. Le lecteur qui voudra approfondir les commandes Hayes (y compris pour l'envoi de fax) pourra consulter ce document de référence : <http://cjovent.free.fr/files/at.pdf>. En revanche, de son côté, le modem termine une émission d'octets par une nouvelle ligne stricto sensu, c'est-à-dire la séquence `\r\n` soit un retour chariot et un saut de ligne. Les exemples de cet article sont garantis par votre serveur comme fonctionnant avec tout modem externe sous Windows et Linux. Ils fonctionneront peut être avec un Modem interne ou "Winmodem" sous Windows et ne fonctionneront pas dans ce cas sous Linux. Enfin le lecteur qui voudrait travailler avec un autre type de périphériques, un lecteur de code barre par exemple, ne devrait éprouver aucune difficulté à adapter les exemples selon ses besoins, les principes de programmation demeurant les mêmes.

2 Installation de l'API Java Communications

L'archive doit être décompressée dans un endroit à votre convenance. Les classes Java, concrètement l'archive `comm.jar`, devra être pointée par le `CLASSPATH`. En outre un fichier de configuration, `javacomm.properties` devra lui aussi être pointé par le `CLASSPATH`. Étrangement ce fichier se cache dans le sous répertoire `docs` de l'archive Linux. Enfin les librairies natives, soit le fichier `.dll` sous Windows et les deux fichiers `.so` sous Linux, devront être pointés par le `PATH` système ou par la variable d'environnement `LD_LIBRARY_PATH` sous Windows et Linux respectivement. Autre possibilité, lancer la JVM ainsi :

```
java -Djava.library.path=le_repertoire_des_libs // etc, etc
```

Le lecteur désirant travailler avec les projets Netbeans fournis sur le Cd-Rom ou le site, devra les adapter à sa propre configuration.

3 Identifier les ports

Au niveau du système le port série est vu comme un fichier, ce qui n'étonnera personne sous Linux mais qui est un peu moins habituel sous Windows. Si l'on programme avec un langage natif, C par exemple, on ouvre ce fichier, puis les octets écrits ou lus dans ce fichier sont autant d'octets envoyés vers, ou reçus du port série. Sous Java le mécanisme est évidemment semblable. Que ce soit en C ou en Java, il faut, sous Linux, avoir le droit d'accéder au descripteur de périphérique, en principe `/dev/ttySO` comme premier port. Par défaut ces droits ne sont accordés qu'à root. Vous devrez donc travailler sous ce compte, ou bien modifier les droits sur `/dev/ttySO`. Une bonne approche est sans doute de créer un groupe utilisateur pour cela. Accorder alors les droits de lecture/écriture au groupe et intégrer à ce groupe tout utilisateur autorisé à accéder au port série. Sous Windows il n'y a rien de particulier à faire. Cependant le nom du fichier n'est pas le même que sous Linux. Sous Windows c'est normalement COM1: comme premier port, mais bizarrement, c'est avec COM1 que l'API Java travaille. Le code devant se comporter à l'identique sur chaque plate-forme, l'API fournit logiquement un moyen d'énumérer les ports, séries ou parallèles, disponibles. Voici un morceau de code qui énumère les ports tout en filtrant les ports série, malheureusement sans garantir le bon ordre des ports énumérés :

```
import java.util.Enumeration;
import javax.comm.*;

public class EnumPort {
    public EnumPort() {
    }

    public static void main(String[] args) {
        Enumeration portIdentifiers =
            CommPortIdentifier.getPortIdentifiers();

        while (portIdentifiers.hasMoreElements()) {
            CommPortIdentifier pid =
                (CommPortIdentifier)portIdentifiers.nextElement();
            if(pid.getPortType() == CommPortIdentifier.PORT_SERIAL) {
                System.out.println(pid.getName());
            }
        }
    }
}
```

Avec ma machine dual-boot, cet exemple imprime :

```
/dev/ttySO
/dev/ttyS1
```

et

```
COM1
COM3
```

sous Linux et Windows respectivement. Mais sur cette machine il n'y a qu'un seul port série. Ainsi sous Linux `/dev/ttyS1` n'est qu'un descripteur de périphérique, c'est-à-dire un fichier système sans aucune correspondance physique. Tenter d'ouvrir ce port serait bien sûr voué à

l'échec. Une application évoluée proposerait ici une boîte de dialogue afin que l'utilisateur choisisse avec quel port travailler. Pour nos exemples j'ai fait le choix primitif de travailler systématiquement avec le premier port (`/dev/ttySO` ou COM1). Il n'y a alors qu'à identifier la plate-forme. Sous Windows une JVM peut renvoyer 'Windows', 'NT', 'Windows XP', 'Windows Vista', etc. comme nom de plate-forme hôte. Pour cette raison, les exemples adoptent l'attitude simple et paresseuse de tester si l'on est ou non sous la plate-forme 'Linux' :

4 Un exemple instinctif, mais qui ne fonctionne pas

Ouvrir et configurer le port série ne présente pas de difficulté en soi. Nous savons que le port série est vu comme un fichier, et que nous pouvons lire et écrire des octets dedans. Nous pouvons donc essayer d'écrire un petit programme d'essai pour nous faire la main. Ça ne fonctionnera pas, c'est ce qui fait tout l'intérêt de l'essai : Ce programme, le voici (UltraBasic.java sur le Cd-rom ou le site) :

```
import java.io.*;
import java.util.*;
import javax.comm.*;

public class UltraBasic {

    // code partiel
    // cf. Cd-rom pour le code complet

    // Ouverture du port
    try {
        sp = (SerialPort) cpid.open( "Me", 5000);
    } catch (PortInUseException e) {
        System.out.println(
            "Ouverture impossible, le port est occupé: " + e);
        System.exit(1);
    }

    // Configuration du port
    try {
        sp.setSerialPortParams(9600,
            SerialPort.DATABITS_8,
            SerialPort.STOPBITS_1,
            SerialPort.PARITY_NONE);
    } catch (UnsupportedCommOperationException e) {
        System.out.println(
            "Impossible de configurer le port" + e);
        sp.close();
        System.exit(1);
    }

    BufferedReader serial_in = null;
    PrintStream serial_out = null;
    try {
        serial_in = new BufferedReader(
            new InputStreamReader(sp.getInputStream()));
        serial_out = new PrintStream(
            sp.getOutputStream(), true, "ISO-8859-1");
        serial_out.println("AT");
        serial_out.println("\r");
    }
```

```

System.out.println(serial_in.readLine());
serial_in.readLine(); // Une ligne à la trappe
System.out.println(serial_in.readLine());
} catch (IOException e) {
    System.err.println("Erreur d'entree/sortie: arret");
    System.exit(1);
}
}
}

```

Nous ne commentons pas l'ouverture du port. Sa configuration, par un appel à `setSerialPortParams` ne fait rien d'autre que renouveler la configuration par défaut. Rien à signaler ici. Des essais ont montré qu'une configuration plus subtile ne pose pas de problème en soi. Le point intéressant est que, une fois le port ouvert, nous pouvons obtenir deux flux, instances de `OutputStream` et `InputStream` pour l'écriture et la lecture sur le port respectivement. Par commodité nous décorons le `OutputStream` par un `PrintStream`. Il est très important de remarquer ici que nous demandons l'encodage `iso-8859-1` pour le flux. En effet, un périphérique série est un périphérique caractère, comprendre ASCII, qui ignore tout de Unicode. Or par défaut Java travaille en Unicode, ce qui ne peut convenir. Ensuite nous envoyons la commande Hayes standard AT. A cette commande, tout modém propre sur lui répond par OK suivi d'une nouvelle ligne, MAIS seulement après avoir ré-émis en écho la commande reçue, retour chariot inclus, ainsi qu'une nouvelle ligne. Notre programme doit donc lire trois lignes. La première contient AT et est imprimée, la seconde, vide, passe à la trappe, enfin la troisième contient OK et est imprimée. Notre code rudimentaire fonctionne sous Windows, mais pas sous Linux. Pourquoi ? Cela tient aux différences de gestion du port entre les systèmes. Windows alloue de petits tampons pour le port série qui est donc, si l'on peut dire, un fichier légèrement bufferisé. Dès que le A de la commande est écrit par notre code, le modém le ré-expédie sur le port. Windows le voit, lit l'octet et le place dans ce mini tampon. Et ainsi de suite. Ensuite la lecture de l'`InputStream` est faite depuis ce tampon. Linux ne fonctionne pas du tout ainsi. Aucun tampon. Quand le A est écrit, il est là aussi ré-expédié, mais doit être lu. Notre code ne lit pas immédiatement, le port s'embouteille puis bloque irrémédiablement. Ne pas se laisser tromper par le nom `BufferedReader` de l'exemple. Nous parlons d'absence de tampon au niveau du système, pas de Java. En fait notre programme est vraiment mauvais. Il ne fonctionnera pas non plus avec, par exemple, la commande AT16 qui demande un diagnostic et provoque une sortie pléthorique de la part du modém. Non seulement notre programme devrait savoir combien de lignes sont à lire, ce qui serait une approche bien maladroite, mais sur-tout le mini tampon saturerait et des octets seraient perdus au cours de l'opération. Bien avertis, nous devons revoir notre copie.

5 Une règle d'or, la programmation asynchrone.

Pour travailler dans des conditions idéales un programme doit lire et écrire les octets un à un, et ceci dans un thread dédié. Ainsi font par exemple les émulateurs de terminal `minicom` sous Linux et `HyperTerminal` sous Windows. Notre exemple suivant adopte par souci de concision, une approche intermédiaire. L'écriture est faite non octet par octet, mais ligne par ligne et sans thread dédié. D'ailleurs, sauf erreur de votre serviteur, il n'est pas possible de lire un seul caractère sur la

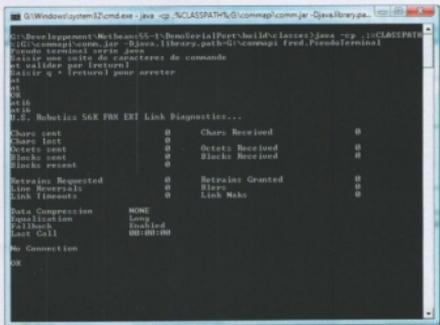
console en Java, mais toujours une ligne. Voilà pourquoi notre exemple est baptisé `PseudoTerminal.java` et non `Terminal.java` :) Par contre rien ne s'opposerait à l'écriture d'un vrai terminal en Swing. Dans notre exemple, toutefois, la lecture est faite, et c'est incontournable, octet par octet dans un thread dédié. Voici le code de ce thread :

```

private class ThreadReader extends Thread {
    public void run() {
        byte[] buffer = new byte[1];
        boolean fini = false;
        try {
            while(!thread_fini) {
                serial_in.read(buffer, 0, 1);
                if(buffer[0] == '\n')
                    continue;
                System.out.print(char[]buffer[0]);
            }
        } catch (IOException ex) {
            // muselée... ATTENTION
        }
        System.out.println("Fin du thread de lecture");
    }
}

```

Rien d'exceptionnel là dedans. Remarquons simplement que nous sautons les retours chariots pour éviter les lignes blanches intempestives. Notons aussi le musèlement d'une exception, acceptable dans cette démonstration, mais qui ne le serait pas dans un programme réel. Ce programme fonctionne parfaitement et vous pouvez y essayer les commandes Hayes qu'il vous plaira.



6 Vous avez dit bizarre ?

Ce programme fonctionne parfaitement ? Non, sous Linux (`BuggyPseudoTerminal.java` dans l'archive Linux) il plante royalement, dès l'appel à `setSerialPortParams` qui lève une exception qui n'est même pas celle spécifiée par la documentation. Dans `BuggyPseudoTerminal` l'appel litigieux est situé dans le constructeur de la classe principale, ce qui est légal, et aussi incroyable que cela paraisse c'est l'unique cause du plantage. Nous avons là un somptueux bug de la JVM et/ou de l'API!

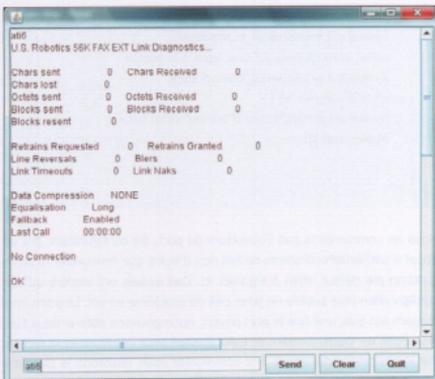
Pour contourner cela, il suffit de faire la configuration du port après la construction de la classe principale, ce que font tous nos autres exemples Linux.

7 Comme c'est bizarre

Nous n'en avons pas fini avec les surprises. Quand le modem émet la séquence `\r\n`, c'est bien cette séquence qui arrive dans le thread de lecture sous Windows. Sous Linux il en va autrement. L'API croit bon de remplacer `\r` par `\n`, donc c'est la séquence `\n\n` qui arrive dans le thread de lecture. L'API procède de même pour les simples retours chariot (`\r`) dans les échos de commandes. Voilà qui explique l'apparition des lignes vides intempestives sous Linux. On voit que l'écriture d'un programme au comportement identique sur les deux plates-formes n'est pas facilitée...

8 Un pseudo terminal sous Swing

Le lecteur trouvera encore deux exemples sur le Cd-Rom ou le site. Le premier exemple, `PseudoTerminalGui.java` utilise la méthode `append` d'un `JTextArea` pour afficher les octets au fur et à mesure de leur lecture. Ceci ne fonctionne correctement que parce que cette méthode est synchronisée. Si l'on voulait travailler avec une méthode non synchronisée d'un composant Swing quelconque, il faudrait procéder de la façon expliquée dans "La programmation multithread avec Java et Swing" de Programmez! 64. Dans le contexte du port série, cette approche est lourde et inefficace. Heureusement, il existe une autre voie, semblable à celle proposée par la classe `SerialPort` de .Net. Dans cette voie, on enregistre une classe quelconque comme écouteur d'événements, puis on demande à l'API de générer ces événements,



par exemple lorsqu'un octet arrive sous le port, ou même d'autres événements relatifs au port série. L'intérêt est que c'est l'API qui se charge alors d'invoquer correctement notre gestionnaire d'événement qui du coup est exécuté dans la pompe à événements du thread AWT. Bref, notre gestionnaire d'événements et Swing sont maintenant en harmonie et nous pouvons invoquer des méthodes Swing non synchronisées. Le dernier exemple, `PseudoTerminalGuiEvent.java` illustre comment organiser son code dans ce contexte.

■ Frédéric Mazué - fmazue@programmez.com

L'INFORMATION PERMANENTE

Programmez!

LA REVUE DU DÉVELOPPEMENT

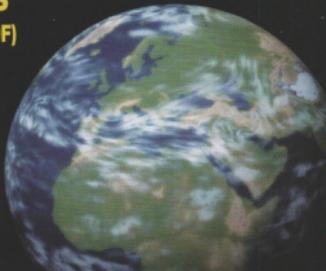
✓ **Abonnez-vous**
au format NUMÉRIQUE (PDF)

Tarif unique pour

LE MONDE ÉNTIER

30 €

(abonnement exclusivement en ligne
www.programmez.com)



✓ **3 NUMÉROS GRATUITS**

Faites des économies, abonnez-vous !

Exemple : en France, vous gagnez 20 euros, soit plus de 3 numéros !

✓ **ÉTUDIANTS**

Tarif spécial **39€**

(France métropolitaine exclusivement)

Nouveau PROGRAMMEZ HORS SÉRIE .NET

Abonnement 1 an : 5 numéros : 20€

COUPON D'ABONNEMENT page 63 de ce numéro ou www.programmez.com

Le langage Dada révélé

Secret jusqu'ici jalousement gardé par les entreprises de développement qui ont une encolure d'avance, le langage Dada est le résultat d'une habile fusion des langages D et Ada. Programmez! vous dévoile son omniprésence insoupçonnée.

En 1974, le département de la défense américain fit un constat hallucinant: pour toutes ses applications informatiques, pas moins de 500 langages étaient utilisés. Un cahier des charges fut alors établi, en vue de la création d'un seul langage pour tout accomplir : Ada était né. De nos jours Ada est tombé en désuétude. Officiellement du moins. Le support du multithreading d'Ada et sa polyvalence restent grandement appréciés. Simplement, mais dans le plus grand secret, le langage s'est vu doté du ramasse-miettes du langage D. Ce nouveau langage, baptisé dès lors Dada, est employé partout, ainsi que l'a révélé notre enquête, qui a débuté lorsque nous nous sommes aperçus que c'est lui qui est employé pour la programmation de la configuration Jumperfree, autrement dit sans cavalier, des nouvelles cartes mères pour processeurs multi-coeurs. Ceci permet un overclocking bestial et programmable du processeur, pour lui secouer les puces et le faire galoper à bride abattue. Dans ce contexte le ramasse-miettes est utilisé pour les alimentations BIOS des pools de threads. Néanmoins un bug malencontreux a pour effet de connecter les cartes son et réseau, ce qui oblige certains systèmes d'exploitation commerciaux, henni soit qui mal y pense, à faire LAN pour avoir du son.

Une technologie omniprésente

Sans que l'on s'en doute, Dada est partout. Celui pour qui la programmation est un hobby l'utilise sans le savoir. Il est de presque tous les systèmes embarqués. Par exemple, c'est du code Dada qui pilote les lecteurs MP3 équi-

panant les Ford Mustang. C'est le langage officiel au plus haut niveau de l'état. Ainsi c'est une routine Dada qui, dans les applications statistiques officielles, assombrir les espaces entre des graphiques trop rapprochés. Récemment notre président, devant le dangereux rapprochement des courbes représentant, l'une en haut la vertigineuse descendante de la richesse du pays, l'autre en bas l'affolante montée de la pauvreté, a commenté ainsi, avec l'art de la cabriole qui est le sien: "le petit monte, l'écart là brunit". On trouve encore Dada dans de nombreux recoins de Windows. Ainsi ce sont des routines Dada qui font des fichiers CAB une assise solide pour les cases à cocher ou qui gèrent le curseur. C'est évidemment le langage de choix pour bon nombre de drivers. Par exemple dans les pilotes de claviers, il assure les retours chariots, dans les pilotes de cartes graphiques, il gère l'animation des Shoot'Em Up totalement bourrins et dans les pilotes de cartes audio il s'occupe de la canne à son.

Un harnais de sécurité

Dada c'est le langage de la sécurité. Avec lui, rien de plus naturel que de concevoir des applications multi-tiercées concurrentes. Dans les anti-virus ce sont des routines écrites en Dada qui protègent contre toutes tentatives de sabotage. Cependant, une carence du langage le rend totalement inefficace contre le moindre cheval de Troie. Dans les centrales nucléaires françaises, c'est encore lui qui pilote les systèmes de sécurité. Son rôle y est de détecter, renifler, sentir les Curies et de s'activer en

conséquence. La cryptographie est son domaine par excellence. Ce langage de l'hennis n'a pas son pareil pour encoder expressions et signes cabalistiques.

Un langage de puissance

Dada n'étant pas Pégase, ce n'est pas non plus un langage Open Source, forcément. Toutefois il est à la portée de toutes les bourses, pour peu que l'on soit capable de rester en selle, car l'acquisition d'une seule licence permet d'équiper 736 postes de développement. Les gens au courant de cela disent généralement qu'un Dada vaut 736 boîtes.

Un dressage délicat

Pour puissant qu'il soit, coder en Dada n'est pas chose immédiate et il se manie parfois difficilement. Il faut d'abord l'apprivoiser et c'est pourquoi il a longtemps été réservé aux programmeurs blanchis sous le harnais. Le monde du Web s'en souvient cruellement. C'est un module écrit en Dada par un développeur bourrin qui avait pendant un temps rendu le serveur Bronco Apache totalement indomptable et inutilisable, avec pour désastreux effets de bord la destruction systématique et générale de tous clusters. C'est pourquoi le monde de l'Internet préfère une version simplifiée du langage Dada, outil qui a notamment servi pour l'écriture du célèbre logiciel eMule. Cependant, en dépit de ses rebuffades, bien apprivoisé, ce langage reste le meilleur ami du fêru d'informatique.

■ Frédéric Mazué et Al Chanu

BOUTIQUE BOUTIQUE BOUTIQUE

Achetez les magazines, les articles en PDF

et abonnez-vous en ligne

www.programmez.com



Maîtriser l'acquisition d'images sous Windows Vista

Sous le nom de Picture Acquisition SDK et comme alternative à l'ardue API WIA, Windows Vista offre une API à la fois simple, souple et puissante pour l'acquisition d'images. Nous apprenons à la mettre en œuvre en C++.

Nous entamons aujourd'hui une série d'articles sur l'acquisition, la capture d'images et de vidéos sous Windows Vista. Les exemples sur lesquels s'appuient ces articles sont tirés et adaptés d'une application développée par votre serveurur. Cette application se charge d'une part de faire des acquisitions d'images depuis différents appareils et d'autre part de faire des captures depuis des webcams à intervalles réguliers, puis expédie les images capturées dans des e-mails. Nous traitons aujourd'hui l'acquisition d'images. Avant Vista, nous disposions essentiellement des API Still Image et WIA pour faire ce travail. Ces API sont d'un usage malcommode, WIA notamment, dont le rôle est plus d'être employée par des scripts supportant Automation qu'autre chose.

1 Vue d'ensemble de la Picture Acquisition SDK

Avec celle-ci les choses sont plus aisées. La programmation COM est toujours présente mais plus conviviale. La SDK offre des boîtes de dialogue préfabriquées facilitant la création d'interfaces utilisateur. L'implémentation par défaut des interfaces COM copie les images des appareils vers le répertoire Images de l'utilisateur Vista. Il est possible de gérer l'arborescence de stockage des données sous ce répertoire. Après acquisition, les photos sont immédiatement présentées par la galerie de photo Windows. Celle-ci conserve des informations sur les images importées ce qui permet d'éviter des acquisitions multiples d'une même image. Enfin il est possible, toujours avec l'implémentation par défaut, de court-circuiter la galerie d'images et de stocker les images dans le répertoire de notre choix.

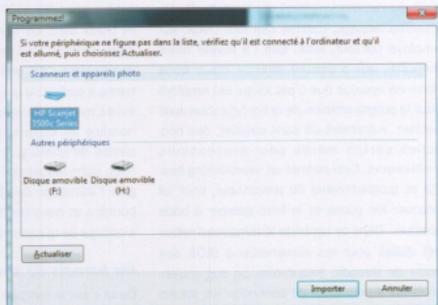
2 La programmation COM en C++

Nous avons très souvent abordé les principes de COM et de sa programmation dans les pages de Programmez!, et récemment dans Programmez! 106 à propos de l'API Vista Contact. Nous n'y reviendrons pas. Signalons simplement, et comme d'habitude, que nos exemples manipulent les interfaces COM au moyen d'un pointeur intelligent conçu à cet effet. Grâce à cela nous tirons avantage de l'idiome C++: acquisition de ressources par initialisation. Cela permet de rendre le code plus concis et aère sa logique. Dans le cas où notre fonction auxiliaire TestOk détecte une erreur, une exception est levée, ce qui a pour effet de libérer toutes les interfaces et autres ressources COM. Le Lecteur qui souhaiterait travailler à partir de nos exemples ou simplement utiliser nos classes utilitaires COM est invité à bien remarquer que tous les exemples commencent par instancier la classe COMInitializer. La présence de cette instance assure, comme le nom le suggère, l'initialisation de la plomberie COM pour l'application. Tous nos exemples ont été écrits avec Visual Studio 2005 et testés avec succès sur Visual Studio 2008 et ne devraient poser aucun problème d'adaptation à un autre

outil de développement pourvu que l'on dispose de la SDK Windows Vista librement téléchargeable sur le site de Microsoft.

3 Détecter les périphériques supportant l'acquisition

Pour cela, l'API fournit une interface COM qui ouvre une boîte de dialogue permettant à l'utilisateur de sélectionner un appareil. Le code ci-dessous montre comment ouvrir cette boîte de dialogue et récupérer l'identificateur du périphérique sélectionné.



Une boîte de dialogue exposant les périphériques supportant l'acquisition.

```
#include <iostream>
#include <string>

#include "COMUtils.h"
#include <PhotoAcquire.h>

#pragma comment(lib, "PhotoAcquireUID.lib")

int main()
{
    try
    {
        HRESULT hr;
        COMInitializer ci;
        SmartCOMPtr<IPhotoAcquireDeviceSelectionDialog> pdsd;
        BSTR name = 0;
        DEVICE_SELECTION_DEVICE_TYPE dsdt;

        hr = ::CoCreateInstance(
            CLSID_PhotoAcquireDeviceSelectionDialog,
```

```

        NULL,
        CLSCTX_INPROC_SERVER,
        IID_IPhotoAcquireDeviceSelectionDialog,
        reinterpret_cast<void**>(&padsd));
COMHelper::TestOk(hr);

hr = padsd->SetTitle(L"Programmez!");
COMHelper::TestOk(hr);

hr = padsd->DoModal(NULL,
    /*DSF_WPD_DEVICES | DSF_WIA_CAMERAS
    | DSF_WIA_SCANNERS | DSF_STI_DEVICES
    | DSF_FS_DEVICES | DSF_DV_DEVICES
    | */
    DSF_ALL_DEVICES | DSF_SHOW_OFFLINE,
    &name,
    &dsdl);
// Si tout se passe bien, mais que l'utilisateur ne
// sélectionne aucun appareil, on arrête
if(hr == E_ABORT)
    return EXIT_SUCCESS;
// sinon on continue
COMHelper::TestOk(hr);
std::wcout << "Appareil selectionne, " << name << std::endl;
std::cout << "Type: ";
std::string device_type;
switch(dsdl)
{
    case DST_UNKOWN_DEVICE:
        device_type = "inconnu";
        break;
    case DST_WPD_DEVICE:
        device_type = "Windows Portable Device";
        break;
    case DST_WIA_DEVICE:
        device_type = "Windows Image Acquisition";
        break;
    case DST_STI_DEVICE:
        device_type = "Still Image Architecture";
        break;
    case DSF_TWAIN_DEVICE:
        device_type = "Twain Device (non supporte)";
        break;
    case DST_FS_DEVICE:
        device_type = "Disque amovible";
        break;
    default:
        device_type = "?????";
        break;
}
std::cout << device_type << std::endl;
}
catch(COMException ce)
{
    std::cout << ce.reason() << std::endl;
    return EXIT_FAILURE;
}

```

```

}
return EXIT_SUCCESS;
}

```

Comme on le voit dans le code, il est possible de sélectionner le type des périphériques affichés. Notre exemple demande tous les périphériques. Au moment où l'illustration ci-contre a été réalisée, cinq périphériques USB étaient connectés à la machine : un graveur de DVD, une webcam, un scanner, une clé USB, et un appareil photo numérique. Sans compter bien sûr des disques dur classiques (non USB). Il est intéressant de noter que la webcam n'apparaît pas dans le dialogue. En effet ce périphérique n'est pas considéré comme un périphérique d'acquisition d'image fixe. Le scanner, qui apparaît dans le dialogue entre bien dans cette catégorie. Cependant l'acquisition avec lui n'est pas immédiate comme elle peut l'être avec un appareil photo. Nos exemples ne fonctionneront pas avec le scanner qui mérite un article à part. La boîte de dialogue montre encore deux périphériques de type disque amovible sous les lettres F: et H: ce qui correspond à la clé USB et à l'appareil photo respectivement. Ceci amène plusieurs remarques intéressantes : D'abord un appareil photo n'est rien de plus qu'un disque amovible pour le PC (périphérique type DST_FS_DEVICE). Cela veut dire que rien n'empêche de traiter l'acquisition de zéro en analysant et traitant le contenu de ce disque. L'API que nous utilisons aujourd'hui est un cadre qui facilite le travail, mais rien de plus. Ensuite nous concluons qu'il est possible d'acquiescer des photos d'une vulgaire clé USB. Dans ce cas, l'API explorera la totalité de l'arborescence de la clé et importera tous les fichiers images (.png, .jpeg, etc) qu'elle aura rencontrés. En revanche, les autres périphériques de type disque n'apparaissent pas dans le dialogue, l'API se limitant manifestement (et logiquement) à l'énumération des périphériques USB et répertoriés "Disk", et non "CdRom", sous la clé de registre USBSTORE dont nous parlerons plus loin. Pour bien se convaincre qu'il n'y a pas de différence entre un disque et un appareil photo en ce qui concerne l'acquisition, il suffit de regarder les identifiants retournés par la boîte de dialogue. Pour un scanner, on aura quelque chose comme :

```
{6BDD1FC6-B10F-11D0-BEC7-08002BE2092F}\0000
```

C'est-à-dire essentiellement un GUID identifiant un appareil, tandis que pour la clé et l'appareil photo, on a respectivement F:\ et H:\, C'est-à-dire des lettres de lecteurs. Ce comportement dicté par l'API peut être apprécié ou pas. On peut penser que ce n'est pas faciliter la vie de l'utilisateur que de lui présenter une lettre de lecteur plutôt que le nom de l'appareil. Si l'on veut remédier à cela, on créera une boîte de dialogue personnalisée. Comment faire alors ? La lettre du lecteur doit être connue puisque l'API d'acquisition l'utilise mais nous devons trouver à quel appareil elle est associée. Il est bon également d'être sûr que le périphérique est connecté physiquement et allumé :) Une démarche possible est d'abord d'énumérer les volumes montés (et donc connectés et allumés) avec les API FindFirstVolume, FindNextVolume et FindVolumeClose. Un nom de volume ressemble à quelque chose comme :

```
\\?\\Volume{45484499-69de-11d0-816b-806e6f6e6963}
```

Ces noms de volumes sont autant de clés dans la base de registres, sous :

```
HKEY_LOCAL_MACHINE\SYSTEM\MountedDevices
```

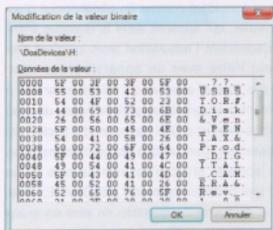
Le terme MountedDevices ne devant pas tromper ici par rapport à ce

que nous venons de dire juste avant. Ici, monté ne veut pas dire connecté mais connu du système, même si déconnecté. Il y a donc a priori plus de clés ici que de volumes énumérés par les API FindFirstVolume et autres. Ensuite on examine les valeurs des clés des volumes énumérés et on les compare avec les clés sous :

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USBSTOR
```

Jusqu'à trouver le nom du périphérique recherché
Ceci fait, on revient à :

```
HKEY_LOCAL_MACHINE\SYSTEM\MountedDevices
```

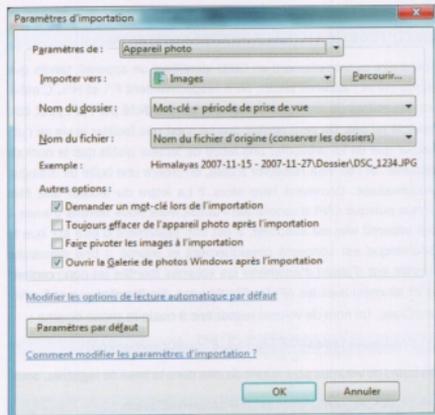


Exemple de lecteur logique associé à un nom de périphérique dans le registre.

Et on recherche quelle clé 'DosDevice' a comme valeur le nom de notre périphérique, ce qui permet d'obtenir la lettre (F:\, H:\ etc.) associée à notre appareil, comme illustré ci-contre) Oui, tout cela est très fastidieux et cela aurait été vraiment mieux que la boîte de dialogue affichée à la fois la lettre et le nom du périphérique...

4 Définir les options d'acquisitions

L'API propose à nouveau une boîte de dialogue pour cela. Le code pour la mettre en œuvre est sans intérêt. Le lecteur le trouvera sur le CD-Rom ou le site. (DemoAcquireOptionsDialog). Remarquons simplement que si la méthode DoModal de l'interface retourne S_FALSE, cela ne signale pas une erreur pour autant, mais simplement que l'utilisateur n'a modifié aucune des options par défaut. Par ailleurs, votre serv



Une boîte de dialogue pour définir les options d'acquisition.

teur n'a pas trouvé comment modifier l'entrée par défaut de la boîte liste déroulante "paramètres de:" ce qui fait que cette boîte de dialogue présente un intérêt assez limité.

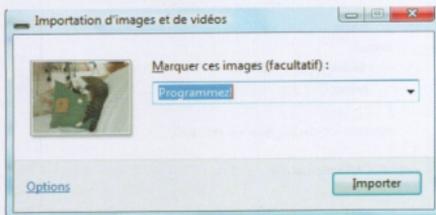
5 Acquérir des images

La marche à suivre est simple. Considérant l'identifiant du périphérique comme étant connu, on commence par obtenir une instance de l'interface COM IPhotoAcquire qui est d'une certaine façon le point d'entrée de l'API dès qu'on aborde les choses sérieuses. Avec la méthode CreatePhotoSource de cette interface et l'identifiant de périphérique, on obtient une instance de l'interface IPhotoAcquireSource qui permet d'accéder aux données au niveau le plus bas. Ainsi c'est avec cette interface que l'on peut énumérer les objets (Items) à acquérir ou encore définir si ceux-ci doivent ou non être effacés du périphérique après l'acquisition. Voici un extrait de l'exemple DemoAcquisition qui montre comment énumérer les objets à partir de cette interface :

```
void EnumerateItems(SmartCOMPtr<IPhotoAcquireSource> pas)
{
    HRESULT hr;
    UINT count;

    std::cout << std::endl <<
        "— Enumeration des items —" << std::endl;
    hr = pas->InitializeItemList(TRUE, NULL, &count);
    COMHelper::TestOk(hr);
    std::cout << "Nombre d'items: " << count << std::endl;
    std::cout << std::endl;
    for(UINT i = 0; i < count; i++)
    {
        BSTR name;
        SmartCOMPtr<IPhotoAcquireItem> pai;
        hr = pas->GetItemAt(i, &pai);
        COMHelper::TestOk(hr);
        hr = pai->GetItemName(&name);
        std::wcout << "Item " << i << " : " << name << std::endl;
    }
}
```

Même si l'on ne souhaite pas énumérer les objets, la méthode InitializeItemList doit être invoquée pour que l'acquisition proprement dite puisse être effectuée. Cette acquisition se fait tout simplement avec la méthode Acquire de IPhotoAcquire qui reçoit en argument l'interface IPhotoAcquireSource. L'API ouvre alors des boîtes de dialogue infor-



Une des boîtes de dialogue apparaissant lors de l'acquisition.

mant de la progression des opérations. Finalement, le point le plus important de notre exemple est surtout l'accès et la configuration aux paramètres d'acquisition. L'accès est obtenu par une instance de IPhotoAcquireSettings. Voici un nouvel extrait de notre exemple, qui définit dans quel répertoire les photos seront importées, répertoire enrichi d'un sous-répertoire ou "GroupTag" du nom de Programmez!

```
SmartCOMPtr<IPhotoAcquire> pa;
SmartCOMPtr<IPhotoAcquireSource> pas;
SmartCOMPtr<IPhotoAcquireSettings> past;

// ...

// Obtenir une IPhotoAcquireSource
hr = pa->CreatePhotoSource(DeviceID, &pas);
COMHelper::TestOk(hr);

// Obtenir une IPhotoAcquireSettings
// et imprimer quelques données de configuration
hr = pas->GetPhotoAcquireSettings(&past);
COMHelper::TestOk(hr);
DWORD dwData;
BSTR bstrData;
hr = past->GetFlags(&dwData);
COMHelper::TestOk(hr);
hr = past->GetGroupTag(&bstrData);
COMHelper::TestOk(hr);
std::wcout << " GroupTag: " << bstrData << std::endl;
hr = past->GetOutputFilenameTemplate(&bstrData);
COMHelper::TestOk(hr);
std::wcout << " OutputFilenameTemplate: " << bstrData << std::endl;

// Définir quelques données de configuration et imprimer pour vérifier
std::cout << std::endl << " _____ "
<< std::endl;
hr = past->SetGroupTag(L"Programmez!");
COMHelper::TestOk(hr);

hr = past->SetOutputFilenameTemplate(
L"$(MyPicturesFolder)\$(GroupTag)\$(RelativePathname[\\])"
L"$(OriginalFilename).$(OriginalExtension)");

COMHelper::TestOk(hr);
hr = past->GetGroupTag(&bstrData);
COMHelper::TestOk(hr);
std::wcout << "Nouveau GroupTag: " << bstrData << std::endl;
hr = past->GetOutputFilenameTemplate(&bstrData);
COMHelper::TestOk(hr);
std::wcout << "Nouveau OutputFilenameTemplate: " << bstrData <<
std::endl;
```

Les données de configuration sont lues dans la base de registre. Le locuteur est invité à se tenir à l'écart de la douteuse méthode InitializeFromRegistry de l'interface et dont la documentation nous raconte que la structure des données dans le registre n'est pas encore définie. Ces données existent bel et bien et vous les trouverez sous la clé :

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\
Photo Acquisition\Scanner\OpticalMedia\Camera etc...
```

C'est encore avec cette interface IPhotoAcquireSettings que l'on travaillerait si l'on souhaitait fournir une boîte de dialogue en remplacement de celle affichée par IPhotoAcquireOptionsDialog, ce qui n'est pas très heureux comme nous l'avons dit.

Le terme "mot-clé" dans ce dialogue correspond à "GroupTag" dans les données récupérées par cette interface. La définition du répertoire complet de destination à l'acquisition ressemble à une incantation ésotérique, mais finalement compréhensible après lecture attentive de la documentation. Analysons notre exemple :

```
MyPicturesFolder\$(GroupTag)\$(RelativePathname[\\])$(OriginalFilename).$(OriginalExtension)"
```

Nous n'avons en fait pas là un répertoire, mais un modèle de répertoire, qui dicte à l'API le comportement à adopter. Cela correspondra au répertoire "Images" de l'utilisateur puis au sous-répertoire (Programmez dans l'exemple) défini par GroupTag.

Ensuite viendront d'éventuels sous-répertoires qui seront la réplique de ceux trouvés dans le périphérique, puis viendront les noms de fichiers et leurs extensions également tels que trouvés dans le périphérique. Il est possible de définir d'autres modèles, à partir des informations fournies dans la documentation MSDN.

Une fois l'acquisition effectuée, Windows conserve une trace des images importées dans les données de travail de l'application "Galleries de photos Windows", dans le fichier :

```
G:\Users\fred\AppData\Local\Microsoft\Windows Photo Gallery\
Pictures.p4d
```

Si vous tentez une nouvelle acquisition sans effacer ce fichier auparavant, l'API vous dira qu'il n'y a pas de nouvelles photos à acquérir depuis votre appareil.

6 Une acquisition personnalisée

Le principe est tout simple. Au cours de l'énumération des objets, on demande un Stream, au sens Shell Windows du terme, autrement dit une interface COM sur un fichier. Il suffit alors de lire le contenu de ce fichier et de le copier dans le répertoire de son choix. Plus de modèle de répertoire ici, mais il faut toutefois prendre une précaution : effacer le fichier Pictures.p4d.

En effet l'API l'examine toujours et si vous lancez notre dernier exemple (DemoAcquisitionPerso) immédiatement après l'exemple précédent, vous auriez la surprise d'apprendre qu'il n'y a pas de photo à acquérir. Le code ne présente pas de difficultés particulières, hormis quelques finesses avec l'interface IStream qui code le déplacement du pointeur de fichier avec un LARGE_INTEGER, type qui est en fait une union, définie dans l'en-tête winnt.h. On notera encore que la méthode Read de l'interface retourne aussi bien S_OK que S_FALSE en cas de succès ;) Cette dernière valeur signifiant une fin de fichier atteinte et non une erreur de lecture.



Flash CS3

- ▀ **Difficulté :** ***
- ▀ **Editeur :** O'Reilly
- ▀ **Auteur :** collectif
- ▀ **Prix :** 42 €

La version CS 3 de Flash offre une nouvelle débauche de fonctions, de nouveautés, d'améliorations. Découpé en 23 sections, ce livre de recettes aborde tous les domaines possibles de Flash CS 3 : données, vidéo, audio, animation, la 3D, les animations, les textes, la création de formulaire, etc. Comme dans les autres ouvrages de la collection, les auteurs posent le problème, donnent la solution puis l'expliquent longuement pour l'utilisateur au mieux, avec exemple de code quand cela est nécessaire. Un must pour les flasheurs qui peut aussi intéresser les développeurs web en général voulant en savoir un peu plus.

Visibilité sur le web



- ▀ **Difficulté :** **
- ▀ **Editeur :** Pearson
- ▀ **Auteur :** Shari Thurow
- ▀ **Prix :** 28 €

Aujourd'hui, un site bien conçu, bien développé ne suffit plus ! Il faut autre chose.

Comment fonctionne véritablement le référencement ? Comment utiliser au mieux les techniques des moteurs et robots ? Quelles astuces pour optimiser le référencement ? L'auteur revient sur tous les éléments : mots-clés, les outils payants, les services en ligne, les balises meta ou encore comment créer un site logique et clair pour le visiteur. L'auteur fournit beaucoup de conseils, ce qu'il faut faire et ne pas faire. Indispensable !

Principes des systèmes d'exploitation



- ▀ **Difficulté :** *****
- ▀ **Editeur :** Vuibert
- ▀ **Auteur :** collectif
- ▀ **Prix :** 64 €

Comment fonctionne un OS ? Comment s'articule-t-il ? Quelles sont les familles d'OS ? En

utilisant Java pour démontrer et faire comprendre les principes de l'OS, ce livre propose une approche plutôt originale. Les fondements sont exposés, de la théorie jusqu'aux réalisations effectives qui ont jalonné l'histoire des systèmes d'exploitation : gestion des processus, stockage, entrées/sorties, systèmes distribués, protection, sécurité. Des exercices et exemples spécifiques sont déclinés pour chaque famille de plate-forme actuelle, permettant au lecteur de découvrir le fonctionnement interne de son propre système d'exploitation : Unix, Solaris, Windows, Mac OS X, sans oublier Linux. Seul le prix peut dissuader de l'achat.

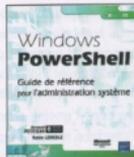
ActionScript 3



- ▀ **Difficulté :** ****
- ▀ **Editeur :** Eyrolles
- ▀ **Auteur :** David Tardiveau
- ▀ **Prix :** 32 €

Le cahier central Flex 3 / AIR de ce numéro vous a peut être donné envie d'aller plus loin. Pourquoi ne pas débuter par ActionScript. L'ActionScript 3, basé sur l'ECMAScript (révision 3 de l'ECMA-262), est un langage totalement orienté objet qui autorise cependant une syntaxe de programmation séquentielle. Très proche du JavaScript, il ressemble également fortement au langage Java. Caractérisé par une syntaxe différente de l'ActionScript 1 et 2, avec de nouvelles classes, propriétés, méthodes et événements, il est plus complexe dans son architecture, mais plus efficace et rapide à maîtriser. L'ouvrage se veut très pratique et on dispose de plus de 180 scripts utilisables pour ses projets. L'ensemble des nouveautés du langage est expliqué.

Windows Powershell



- ▀ **Difficulté :** **
- ▀ **Editeur :** Pearson
- ▀ **Auteur :** collectif
- ▀ **Prix :** 39 €

Powershell fait désormais partie de la panoplie des utilisateurs avancés et administrateurs. Ce puissant shell en ligne de commande permet de piloter, manipuler l'ensemble du système, plus finement qu'avec les interfaces graphiques.

Et la version 2.0 promet encore plus de possibilités, notamment avec une création de ses propres commandes, plus simples. Plutôt complet, le livre aborde les différents aspects de Powershell avec de nombreux codes et exemples, l'intégration avec .net, la sécurité, les objets COM, etc.

ITIL version 3



- ▀ **Difficulté :** **
- ▀ **Editeur :** Editions Eri
- ▀ **Auteur :** Claire Noirault
- ▀ **Prix :** 39 €

Qu'est-ce qu'ITIL ? Bien qu'éloigné du monde des développeurs, le sujet intéresse le chef de projet, le DSI, l'architecte. L'ouvrage tente de démythifier l'approche ITIL et sa raison d'être en entreprise et surtout au niveau du département informatique. On y verra la gestion des capacités, de disponibilité, la continuation des services, la sécurité, etc. On touche là à l'organisation même des projets, des relations entre personnes, la méthode de travail. Le développeur pourra toujours trouver des idées, de bonnes pratiques.

L'analytique nouvel outil stratégique



- ▀ **Difficulté :** ****
- ▀ **Editeur :** Pearson
- ▀ **Auteur :** collectif
- ▀ **Prix :** 31 €

Les données et informations représentent une volumétrie toujours plus importante. Le problème est de savoir comment gérer,

chercher la bonne information, comment en tirer des statistiques, des informations pertinentes ? La réponse est dans l'usage des outils analytiques. En prenant appui sur des exemples très concrets de sociétés et d'éditeurs informatiques, les auteurs décortiquent l'analyse des données, et pourquoi, ces sociétés peuvent en tirer un profit concurrentiel. Passionnant, ce livre nous plonge dans une autre informatique méconnue mais devenue stratégique : l'exploitation et l'utilisation des données... À lire absolument !

WEB 2.0

Ajax en 1 clic avec WEBDEV XII



Vous êtes invité !

11 villes du 27
Mars au 29 Avril
2008

100% technique

Inscrivez-vous
(gratuit) sur
pcsoft.fr

```

ConfirmationCommande : Clic de VisuCodePostal (over)
// Propose automatiquement les villes correspondantes
// la liste est remplie avec la liste des villes correspondantes
// Suppression de l'ancien contenu de la liste
[listeSupprimeTout(VILLES)]
// Recherche des villes (paramétrage de la requête et exécution)
REQ_RechercheVilles.pCodePostal = CodePostal
[RExécuteRequête(REQ_RechercheVilles)]
// Ajout des villes correspondantes
POUR CHAQUE VILLE
    Ajouter(VILLE)
    END POUR
ConfirmationCommande : Clic de VisuCodePostal (over)

```

Ici Ajax est désactivé

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

```

ConfirmationCommande : Clic de VisuCodePostal (over)
// Propose automatiquement les villes correspondantes
// la liste est remplie avec la liste des villes correspondantes
// Suppression de l'ancien contenu de la liste
[listeSupprimeTout(VILLES)]
// Recherche des villes (paramétrage de la requête et exécution)
REQ_RechercheVilles.pCodePostal = CodePostal
[RExécuteRequête(REQ_RechercheVilles)]
// Ajout des villes correspondantes
POUR CHAQUE VILLE
    Ajouter(VILLE)
    END POUR
ConfirmationCommande : Clic de VisuCodePostal (over)

```

Un clic, et Ajax est actif

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 d'administration...: tout est inclus, en français.

Vous aussi réalisez vos sites WEB 2.0 10 fois plus vite... avec WEBDEV XII
(logiciel professionnel, dossier gratuit sur simple demande)



Un des nombreux
exemples livrés avec
WEBDEV : portail
Intranet **Honolulu**,
à télécharger gratuitement
(sur pcsoft.fr).

www.pcsoft.fr info@pcsoft.fr
Dossier technique et témoignages sur simple demande
Tél Province 04.67.032.032. Tél Paris 01.48.01.48.88

UN CODE MULTI-PLATEFORME : WINDOWS, WEB, MOBILE

WEBDEV



Fournisseur Officiel de la
Préparation Olympique

Votre potentiel, notre passion.™

Microsoft

Microsoft
Visual Studio

SATISFAIRE CHACUN, SANS COMPROMIS.
VISUAL STUDIO 2008.



Votre défi : concevoir de meilleures applications pour la bureautique et le web dans les meilleurs délais et presque sans effort.

Votre arme : Visual Studio 2008. Pour gagner en temps, en expérience utilisateur et en qualité, tout simplement. Plus d'informations sur www.relevetouslesdefis.com