

Se préparer à  Windows® 7

PROGRAMMEZ!

Le magazine du développement

PROgrammez!

www.programmez.com

mensuel n°117 - mars 2009



Les **25** erreurs de programmation :
le rapport qui dit tout !

Comment hacker son code ?

Les fonctions de sécurité de .Net

Base de données

- ✓ Partitionnement sous MySQL
- ✓ Guide de survie sur Hibernate
- ✓ Quelles évolutions dans les SGBD ?



Bien
débuter en
PHP

Comment
les éditeurs
séduisent
les étudiants ?



C++

CUDA : optimiser
les applications
avec le GPU

Windows

Manipuler
le Speech Server

Jeux

XNA :
la machine
à jeux ! 2e partie

Multicore

Traquer les bugs

M 04319 - 117 - F: 5,95 €



Nouvelle version **14**

501

NOUVEAUTÉS

Réussissez tous vos projets
avec l'outil de développement élu
le plus productif du marché*.

UN CODE MULTI-PLATEFORMES :

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

VERSION
EXPRESS
GRATUITE
Téléchargez-la !

Ouverture

Robustesse

Pérennité

Réduction des coûts

Rentabilité

Adequation aux besoins

Gestion du changement

Sécurité

14 Nouveau:

Mashup

Lien Google

Lien Salesforce

HyperFileSQL : full
text

DataBinding

Nouveaux
graphiques

Nouvelles tables

Robot de moni-
ring & surveillance

Accès Natif
PostgreSQL

Lien Silverlight 2
et Flex

PHP 5

214 Nouveautés
fonctionnelles

120 Nouvelles
fonctions
WLangage

62 Nouvelles
fonctions Java

32 Nouvelles
fonctions PHP

101 Nouvelles
fonctions LINUX

www.pcsoft.fr

DEMANDEZ LE DOSSIER GRATUIT : 244 pages + DVD
+ Version Express incluse + 112 Témoignages.
Tél: 04.67.032.032 ou 01.48.01.48.88 info@pcsoft.fr

Fournisseur Officiel de la
Préparation Olympique

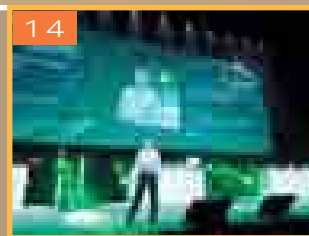


sommaire

\\ actus	
L'actualité en bref	8
Agenda	8

\\ chronique	
La machine vivante et la bio-informatique	12

\\ événement	
TechDays 2009 : objectif .Net 4 et Windows 7	14
Windows 7: il faut s'y préparer dès maintenant !	16



\\ gros plan SGDB	
Base de données : optimiser, sécuriser, relier	20
Les mécanismes de sécurité dans les bases de données	21
Hibernate : guide de survie	26
Partitionnement avec MySQL 5.1	30
Intégration de données avec SQL Server Integration Services	34
Données (géo)spatiales dans SQL Server 2008	36



\\ dossier :	
Hacking, sécuriser votre code	38
Hack your code	39
La sécurité appliquée à .Net	44
Les 25 erreurs de programmation les plus fréquentes	48
Les réactions	52



\\ carrière	
Comment les éditeurs séduisent les étudiants et les écoles	54
Le co-voiturage en temps réel	58



\\ code	
Maîtriser les fondamentaux de PHP ! 1re partie	60
Débuter avec BizTalk Server 2009 (2e partie)	64
Découvrir et mettre en oeuvre CUDA 1re partie	68
Faites parler votre PC avec Microsoft Speech	72
Développez en Ruby On Rails avec 3rd Rail	76
La chasse aux bogues parallèles avec Intel Parallel Inspector	78



\\ temps libre	
Créer un jeu vidéo de A à Z avec XNA (2e partie)	80
Les livres du mois (Spécial sécurité)	82



En complément sur www.programmez.com

CODE

Les sources des articles

EXCLUSIF

Les 25 erreurs de programmation : le rapport complet en français

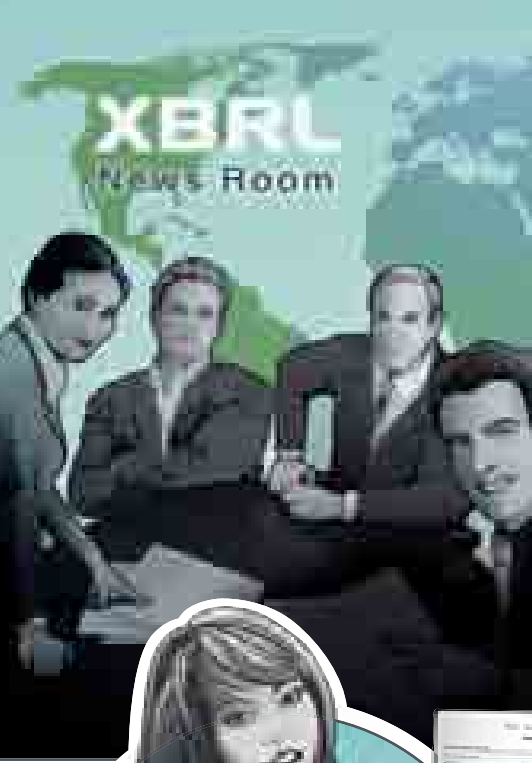
TÉLÉCHARGEMENT

BizTalk 2009
Intel Parallel Studio

Et tous les outils et leurs mises à jour

QUOTIDIEN

Actualité
Forums
Tutoriels
etc.



ALTOVA®
missionkit®

Passez à la norme XBRL grâce aux différents outils Altova®

Découvrez comment Altova MissionKit® 2009, la suite intégrée comprenant des logiciels de langage XML, de bases de données et d'intégration de données, vous offre tous les outils dont vous avez besoin pour travailler en XBRL, et tout cela sans vous ruiner.



Altova MissionKit 2009 inclue des outils intelligents pour éditer, mapper et publier du XBRL:

XMLSpy® – l'éditeur XML leader avec prise en charge du XBRL

- Validation de XBRL et Dimensions
- Edition graphique de la taxonomie XBRL

MapForce® – l'outil graphique de mappage et de conversion des données

- Mappage "any-to-any" de XBRL, bases de données, XML & données Excel 2007
- Génération de rapports d'archivage XBRL par glisser-déposer

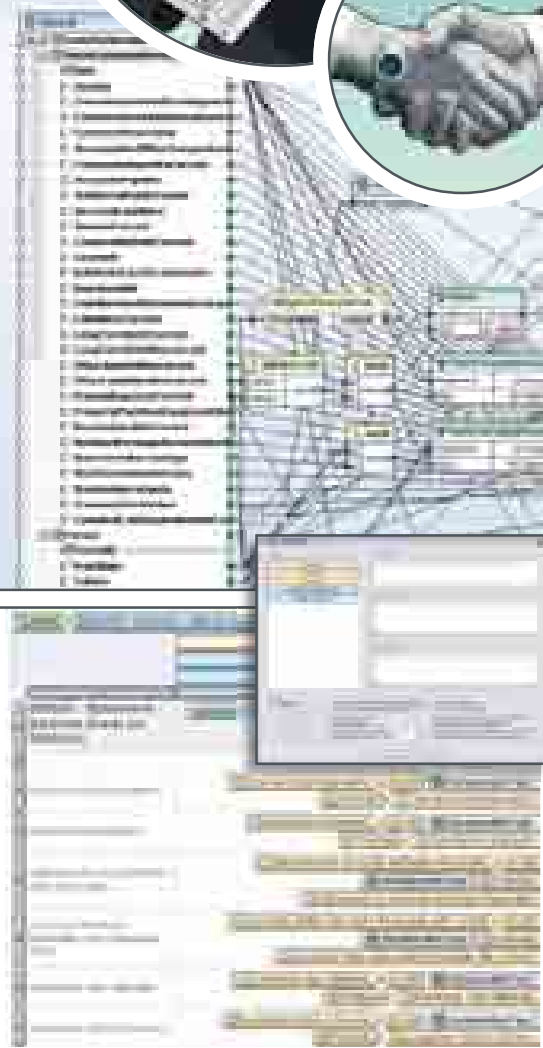
StyleVision® – l'outil de conception de feuilles de style et de publication de rapports

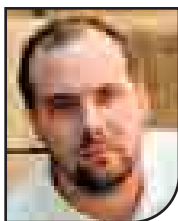
- Publication simultanée de rapports financiers XBRL en HTML, PDF et Word
- Assistant intelligent pour présenter facilement le rendu des données XBRL



Téléchargement gratuit!

Testez le produit avec une version d'essai de 30 jours totalement fonctionnelle, à télécharger sur www.altova.com.





Hacker n'est pas jouer ?

En France le hacking a souvent mauvaise presse. Le " hacker " est le vilain pirate informatique. Dans d'autres pays, le vrai hacker est mieux considéré, voire érigé en " héros ", en maître du code : c'est un art qui s'apprend et se maîtrise avec les années. Dans notre dossier du mois, nous traitons de sécurité, de " hacking ", dans le sens noble du terme. Et nous verrons, grâce à la maîtrise d'une figure montante du " hack " français, IvanleOu, comment hacker son code peut rendre bien des services à un développeur. C'est une des meilleures manières d'éprouver un code, de découvrir des failles parfois vicieuses !

Nous constaterons dans ce numéro mais aussi dans celui du mois suivant, combien notre informatique reste poreuse. Nos logiciels ressemblent à du gruyère, on n'en compte plus les failles. Et pourtant, les surfaces d'attaques sont connues depuis des années et les contre-mesures à prendre apparaissent quasi dérisoires. Cela coule de source ... mais on oublie parfois de mettre un simple contrôle en entrée et en sortie des champs de saisie, ou d'interdire l'injection de code !

Depuis début janvier circule un rapport sur les 25 erreurs de programmation. Dans ce numéro, nous analysons ses conclusions. Mais Programmez ! a été plus loin : nous avons réalisé la traduction intégrale en français que nous vous proposons en ligne, sur www.programmez.com.

A quand une véritable norme internationale sur la qualité et le code sécurisé ? Imaginez que les éditeurs soient légalement responsables des failles de sécurité de leurs applications ? De quoi donner des sueurs froides ! La question mérite une réponse, et rapidement.

Cela nous amène à un sujet hautement sensible : l'informatique prouvée ou " Proof Computing ". Traduction ? Tout " simplement " le fait de disposer de systèmes, de logiciels, de codes pour lesquels on peut prouver le fonctionnement, les résultats. L'informatique devient prédictible. On est capable d'annoncer l'état d'une application dans un temps défini, on sait prédire son comportement. Cela repose sur de puissants mécanismes mathématiques (bases de nos preuves formelles) et sur une nouvelle génération d'outils et de langages à la fois sécurisés et prouvés. Nous disposons déjà partiellement de cet arsenal : ce sont les langages fonctionnels. Demain, tout développeur .Net pourra en faire avec l'intégration de F#.

Il s'agit d'un changement radical de la philosophie du développement. A mon tour de prédire que le Proof Computing s'imposera, lentement mais sûrement, dans les 5 ans. Quelques briques « prouvées » s'insinuent déjà dans notre quotidien.

■ FRANÇOIS TONIC

Rédaction : redaction@programmez.com

Directeur de la Rédaction : Jean Kaminsky

Rédacteur en Chef :

François Tonic - ftonic@programmez.com

Ont collaboré : F. Mazué, C. Remy, IvanleOu, N. Toukis.

Experts : P. Guernonprez, F. Pedro, A. Chen Kuang Piao, O. Lance, H. Darmet, O. Dasini, F. Queudret, Y. Moulin, Bertrand Renotte, S. Haim-Lubczanski, P. Berquet, R. Baduel, G. Vidal.

Crédits photo : © istockphoto

Publicité : Régie publicitaire, K-Now sarl

Pour la publicité uniquement :

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

Editeur : Go-02 sarl, 6 rue Bezout

75014 Paris - diff@programmez.com

Dépôt légal : à parution - Commission

paritaire : 0712K78366 ISSN : 1627-0908

Imprimeur : ETC - 76198 Yvetot

Directeur de la publication : J-C Vaudecrane

Ce numéro comporte un encart libre Solutions Linux sur une partie du tirage, dont les abonnés.

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 : 49 € - Etudiant : 39 € - CEE et Suisse : 55 € Algérie, Maroc, Tunisie : 59 € Canada : 68 € Tom : 83 € - Dom : 66 € Autres pays : nous consulter. **PDF :** 30 € (Monde Entier) souscription en ligne.

L'INFO PERMANENTE
WWW.PROGRAMMEZ.COM



PROCHAIN NUMÉRO

N°118 avril 2009,
parution 30 mars

✓ Dossier Linux / Open Source

Choisir sa distribution de développement, comment développer des applications pour Linux, au cœur des laboratoires Mozilla

✓ Enquête

Votre code est-il légal ?

✓ Développement web

Choisir son framework Ajax

✓ Outil

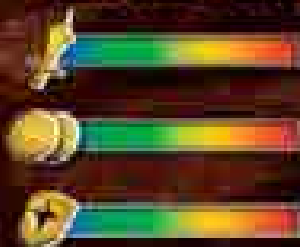
Pourquoi choisir WinDev ?

Votre potentiel, notre passion.[™]
Microsoft®



N'OUBLIEZ JAMAIS : DES POUVOIRS AUSSI PUISSANTS
DOIVENT ÊTRE UTILISÉS POUR FAIRE LE BIEN.

RECEVEZ TOUS LES DÉFIS



Votre défi : concevoir des applications web qui délivrent une expérience riche.

Vos armes : les outils Visual Studio® pour Silverlight avec l'abonnement à MSDN Premium. Tout un monde de nouvelles fonctionnalités pour les designers et les développeurs. Plus d'informations sur releveztouslesdefis.com

■ BLU AGE Build estampillé Ready for IBM Rational Software

L'éditeur français, s'intégrant avec Eclipse, rejoindra Rational Software avec BLU AGE 2009 grâce au label Ready for IBM Rational Software. D'autre part, la suite française obtient le niveau IBM Rational Best Practices Indicator du programme IBM SOA. L'éditeur sera très présent durant la conférence utilisateur Rational fin mai début juin 2009 à Orlando.



■ **Telosys**, Framework Ajax issu de la société Sogeti et disponible sur Object-Web est en version 1.0. Il s'agit d'un Framework de développement utilisant le principe du client / serveur mais dans l'univers Ajax. A découvrir : <http://telosys.objectweb.org/>

■ **Adobe** a reçu mi-février une note "strong positive" dans l'étude Market-Scope sur Ajax et les plates-formes RIA. En tout, une vingtaine de solutions avaient été retenues pour l'étude qui s'appuie sur divers critères dont les produits/services, la stratégie d'offre, la viabilité globale, les ventes et la tarification, la réactivité commerciale et les références et la compréhension du marché.

■ **Microsoft** étend l'interopérabilité de ses outils de virtualisation en prenant un nouvel accord avec le monde open source : Red Hat. Les clients de l'éditeur ayant manifesté un intérêt particulier pour une telle compatibilité ! Concrètement, cela signifie que Red Hat validera le bon fonctionnement de Windows Server dans son environnement virtualisé et que Microsoft fera de même de son côté (Hyper-V).

Outils Bespinn : le nouveau projet de recherche de Mozilla



Courant février, Mozilla Labs dévoilait un nouveau projet : Bespin. Il ne s'agit là que d'un projet de recherche mené par un laboratoire Mozilla. Il ne s'agit nullement d'une application de production ou d'un projet débouchant avec certitude sur quelque chose, seul l'avenir nous le dira. Comme nous l'a précisé Tristan Nitot de Mozilla Europe, les éditeurs en ligne existent depuis longtemps. Le constat de départ de Bespin est : pourquoi ne pas programmer une application web directement sur le web ! Il s'agit surtout d'essayer de définir une nouvelle génération d'éditeur en ligne, en s'appuyant sur des navigateurs modernes et les derniers standards web comme HTML5. La version 0.1 dévoilée est un Framework qui demande à être éprouvé et étendu. Par contre l'un des intérêts de Bespin est pouvoir partager le même fichier de source entre plusieurs utilisateurs... Site : <https://bespin.mozilla.com/>

RIA Moonlight 1.0 : enfin disponible !



Si le prototype du Silverlight open source fut réalisé en à peine 3 semaines, il a fallu aux développeurs Moonlight deux ans pour peaufiner et sortir la pile Silverlight libre en version 1.0. Cette solution permet de développer des applications Silverlight 1.0 fonctionnant sur Linux et de faire fonctionner des applications Silverlight directement sur Linux. Pour le moment, Moonlight supporte Silverlight 1.0, le support de la v2 n'en est encore qu'au début. Un designer XAML est

disponible pour faciliter le développement des applications. Microsoft a fourni un effort en assurant une bonne interopérabilité entre les deux plates-formes. Moonlight est disponible uniquement sur Linux (et Firefox) pour le moment, d'autres systèmes pourraient être supportés. Moonlight 2 supportera Silverlight 2 avec des performances javascript en fortes hausses. A vous de jouer !

Site : <http://www.go-mono.com/moonlight/>

Interface

Comment promouvoir efficacement **OpenScreen** ? Adobe et Nokia donnent une piste en annonçant un fond de 10 millions aux projets Open Screen. Cette aide concerne les applica-

tions et services desktop et mobiles (et autres terminaux) utilisant Flash. Open Screen est un projet important pour Adobe et ses partenaires afin d'assurer le bon affichage de l'interface et du contenu quel que soit le terminal utilisé.

agenda \

MARS 2009

Le 5 Mars, Nantes

Salon les Jeudis.com

dédié au recrutement des ingénieurs et des informaticiens.

<http://www.lesjeudis.com>

Du 10 au 11 mars, Paris Auditorium Paris Centre Marceau

Mobile 2.0

<http://www.lemobile20.fr/2009>

Du 19 au 21 mars, Paris Expo Porte de Versailles, **Salon de l'Internet Pro**

<http://www.salon-internet.fr/pro>

Le 22 mars, Paris Palais Omnisports Paris Bercy, **Salon des formations de l'informatique et des nouvelles technologies.**

<http://www.studyrama.com>

Du 31 Mars au 2 avril, Paris Expo Porte de Versailles

-Hall 2.2 : **Solutions Linux Open Source**, 10e édition de l'événement européen de référence sur le marché Linux et du Logiciel Libre. <http://www.solutionslinux.fr/main.php>

-Hall 8 :

Machine to Machine, pour découvrir l'offre et les innovations des

principaux acteurs du marché MtoM (M2M)

<http://2009.salon-mtom.fr>

-Hall 8 :

RTS Embedded Systems 2009

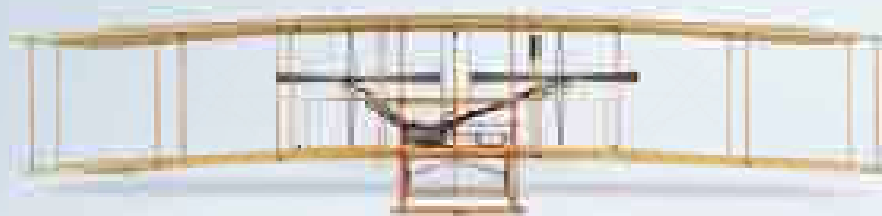
Le salon des solutions informatiques temps-réel et des systèmes embarqués

<http://2009.salon-rtts.fr/>

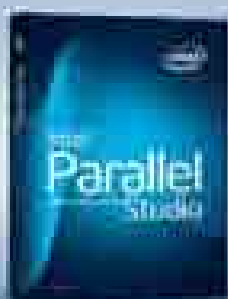
ETRANGER

Du 03 AU 08 Mars 2009, Hanovre, Allemagne, **CEBIT 2009**, le 1er salon mondial des TIC

<http://www.globalfairs.fr/-CeBIT-.html>



ATMOSPHERE. STRATOSPHERE.



FAITES EVOLUER VOTRE CODE.

L'innovation en développement parallèle.

Accélérez vos développements pour le multi-core avec Intel® Parallel Studio. Toute la panoplie d'outils pour changer votre façon de programmer. Fait pour les applications sérielles d'aujourd'hui et les innovations parallèles de demain. Pour en savoir plus et tester la version beta: www.intel.com/go/parallel



■ Tout le monde veut son magasin virtuel pour vendre des applications à l'image d'AppStore d'Apple qui fait le succès de l'iPhone depuis des mois ! Ainsi, **Nokia** et **Microsoft** ont lancé l'offensive à Barcelone mi-février durant le Mobile World Congress. Du côté Google Android, le magasin virtuel accueillera des applications payantes, une grande nouveauté !

■ **Debian 5.0** est disponible depuis le 14 février dernier. Cette nouvelle version majeure, la première depuis 2 ans, inclut un noyau 2.6.26, libc6 2.27, gcc 4.3.2, de nouvelles versions de Gnome et KDE mais aussi d'Apache, Samba, OpenJDK, Xen, etc. Il est possible de l'utiliser via un liveCD.

■ **Sonar 1.6** est disponible. Sonar permet de vérifier et gérer la qualité de son code. Cela inclut les tests unitaires, la couverture du code, le respect des règles de codage, etc. On peut mettre en place des profils utilisateurs selon son rôle dans un projet. Pratique. A télécharger ici : <http://sonar.codehaus.org/>

■ Le **JUG** (Java User Group) de Paris vient de fêter sa première bougie ! Pour l'occasion, de nombreux JUG furent invités. La présentation technique concernait le Framework Wicket et Jazoon 2009, conférence sur Java. La prochaine rencontre est prévue le 10 mars autour de Wicket et le web sémantique. Site : <http://www.parisjug.org>

■ **Glassfish 3.0**, disponible en pré-version, devrait être livré en version finale (si tout va bien) cet été en même temps que Java EE 6. Cette version, très prometteuse, inclut une toute nouvelle architecture modulaire grâce à l'utilisation de OSGi et d'un micro noyau. Site : <https://glassfish.dev.java.net>

■ **Xen** n'est pas mort ! Citrix a dévoilé plusieurs nouveautés autour de la plate-forme de virtualisation : XenDesktop 3 (meilleure prise en charge des contenus et applications multimédia), prise en charge de Citrix HDX par XenDesktop et XenApp. HDX apporte une nouvelle expérience utilisateur plus riche, plus visuelle (voix, vidéo, 3D).

■ Découvrez un éditeur XAML gratuit : **Kaxaml**. Il propose une vue splittée, code et design, et possède une barre d'outils assez complète. Sa qualité est sa petitesse et ses performances. A découvrir sur <http://www.kaxaml.com/>

■ Microsoft Research travaille activement au futur outil de tests unitaires. Les développeurs proposent **PEX** permettant une automatisation très poussée des tests unitaires en s'appuyant sur l'analyse d'exécution de l'application pour couvrir l'application et son code. Site : <http://research.microsoft.com/en-us/projects/pex/>

Mobilité

Android arrive en France

Le premier modèle de téléphone portable embarquant Android est disponible depuis quelques mois excepté en France. C'est en mars que le premier " Google Phone " doit (enfin) arriver dans notre pays ! Il s'agira du modèle HTC G2, le successeur du G1. Il ressemble à un iPhone et perd son clavier physique au profit d'un clavier virtuel ! En attendant, les SSII et consultants se positionnent en France



malgré l'absence de terminaux. Ainsi, Arnaud Farine a créé Expertise @ndroid. La société propose 3 axes : la formation, la conception et le développement et enfin le consulting. Pour A. Farine, il existe une réelle demande sur Android, le modèle de développement, la formation et les conseils de positionnement. " *Je m'attendais à un démarrage (de l'activité) plus fort en début janvier, mais cela ne commence réellement que maintenant, avec une explosion après l'arrivée des premiers téléphones. C'est un peu comparable au iPhone* ", commente A. Farine. Notre expert pointe aussi du doigt que beaucoup de développeurs iPhone se mettent à Android. Il existe déjà plusieurs sociétés sur Android en France. Il ne fait aucun doute que ce marché sera particulièrement actif en 2009.

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

XML

Altova lance MissionKit Tool Suite 2009

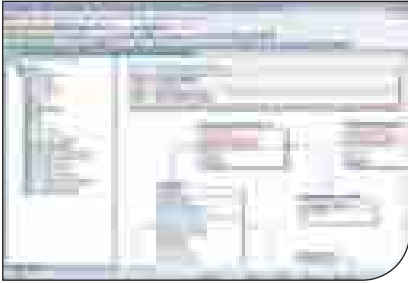
L'éditeur spécialisé dans les outils XML vient de dévoiler la version 2009 de son MissionKit Tool Suite, intégrant les outils XML, de bases de données et UML. Cette nouvelle édition incorpore les standards XBRL (Extensible Business Reporting



Language, standard ouvert pour composer des rapports orientés business) dans l'ensemble des outils de la suite. XMLSpy 2009 propose ainsi un nouveau moteur interne supportant XBRL 2.1 et XBRL Dimensions 1.0. Cela permet notamment de valider une instance de documents XBRL, il possède aussi un éditeur très complet. Pour le mapping XBRL, on passera avec MapForce. Altova supporte désormais les dernières versions des bases de données du marché. D'autre part, l'éditeur annonce aussi l'apparition de HL7 (Health Level Seven) dans MapForce. HL7 est un format d'échange de données médicales ciblant les établissements hospitaliers, les administrations. Ce format est désormais un standard international. Côté modélisation, UModel 2009 facilite le reverse engineering en Java, C# et VB, en étant capable de générer les diagrammes de séquence. Disponible dès maintenant ! site : www.altova.com

Outils

Embarcadero opte pour une offre à la demande



Les éditeurs d'outils se mettent activement aux solutions à la demande. Embarcadero lance le service Embarcadero All-Access. Objectif : donner accès à la demande aux outils nécessaires pour développer, construire, concevoir, exécuter des applications et des systèmes de données ! Et ce, en supportant de

nombreux langages et plates-formes. Cette offre inclut Delphi, JBuilder, C++ Builder, 3rdRail, les outils pour SGBD, soit pas moins de 18 outils ! Par ce biais, l'éditeur veut attirer de nouveaux utilisateurs et adapter l'offre selon les besoins de chacun ! Il s'agit d'un changement important pour l'éditeur, et préfigure ce que d'autres feront dans les prochains mois. Le prix se fera selon le niveau de fonctions voulu. Reste désormais à voir l'accueil de l'offre et le comportement des développeurs. Cela permet aussi de simplifier le modèle de licence (qui devient unique) et de simplifier le déploiement. Dans l'offre All-Access, on dispose aussi de InstantOn qui permet d'utiliser les outils de All Access sans les installer. Ils fonctionneront sur le poste (sans installation) ou directement sur le réseau. Cette offre " spéciale " sera généralisée dans l'offre All Access dans les prochains mois. On a le choix entre 4 niveaux. Prix de départ : 2 250 dollars.

Site : www.embarcadero.com/products/all_access.

■ **O'Reilly et Palm** viennent de rendre disponible sur developer.palm.com le premier chapitre d'un guide de développement de webOS, le système du Palm Pré. Intitulé Palm webOS : développer des applications Javascript avec le Framework Palm Mojo, on pourra enfin voir du concret sur ce système !



■ **Mia-Software** annonce Mia-Insight 3.3 dédié à la cartographie et l'analyse du patrimoine applicatif des entreprises. L'outil utilise une approche MDA et intègre plus finement le langage NSDK. Pour optimiser les performances, l'outil peut travailler sur plusieurs cœurs en parallélisant les tâches.

■ **Quest Software** rend disponible LiteSpeed Engine for Oracle, une solution de sauvegarde et de restauration pour les bases de données Oracle. Il permet surtout d'optimiser la taille des sauvegardes et le temps nécessaire pour les réaliser. Et on dispose de plusieurs niveaux de cryptage des données à sauvegarder.

■ **Red Hat** dévoile Jboss Enterprise Portal Platform 4.3, plate-forme de portail open source. Cette version supporte la JSR-286, le JBoss Portlet Bridge et les outils portlets. Offre visant en premier lieu les applications composites, les mashup, les applications web portails.

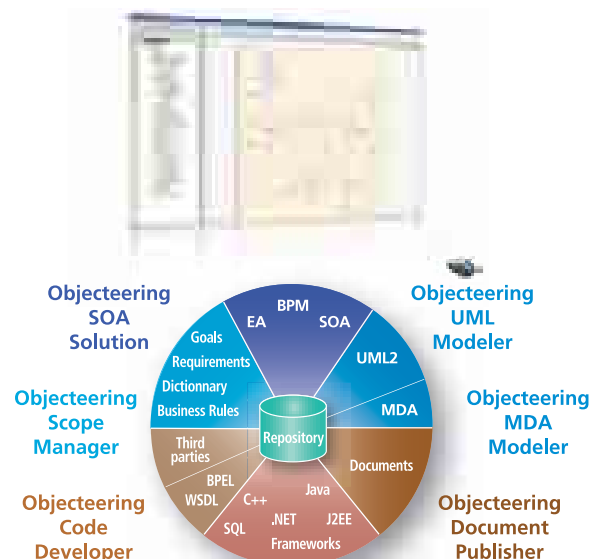
Objecteering

Your projects deserve a tool *

La convergence SOA, UML2, BPMN, EA, pour le développement guidé par le modèle

- outil **intégré** assurant la modélisation complète des applications et systèmes,
- **facile** à utiliser : à chaque acteur, à chaque phase, la représentation dédiée,
- **alignement métier** du SI grâce au référentiel partagé et la traçabilité totale entre toutes les vues et phases,
- haute **productivité** grâce à la génération automatique de la documentation et du code.

* Vos projets méritent un outil.



www.objecteering.com

Tél. : 01 30 12 16 60 - sales@objecteering.com

Objecteering
SOFTWARE

The model-driven development company
Mars 2009 \ PROGRAMMEZ \ 11

La machine vivante et la bio-informatique

Si regarder les actualités vous fait douter de la qualité du code définissant le comportement de l'être humain, cela n'empêche pas d'admirer l'élégance de la machine sous-jacente. C'est elle qui a inspiré les chercheurs, afin de s'attaquer à de nombreux problèmes qui restent inaccessibles aux architectures informatiques classiques.

Si le Graal de l'ordinateur quantique se précise, d'autres possibilités ont été explorées, et notamment celles basées sur le fonctionnement même des machines vivantes.

L'ensemble des données définissant un être vivant se trouvent codées dans les molécules d'ADN (pour plus de détails, voir les séries scientifiques bien connues " Les Experts ", dont la diffusion est recommandée dans toutes les bonnes prisons). Le codage est basé sur quatre molécules (A, T, G, C), sachant que les seuls couples possibles sont A-T, T-A, C-G, G-C. Nous disposons donc d'un moyen de stockage de l'information en base 4. Sans rentrer dans les détails enzymatiques, un MOT ne comprend pas 8, 16 ou 32 bits, mais 20 paires.

Les premières applications

La première application est donc le stockage d'informations à haute densité. La molécule d'ADN étant stable, on peut envisager une durée d'archivage utilisable. Il est également possible de coder l'information sur un brin unique d'ADN. Comme on vient de le voir, deux brins ne pourront se recombiner que si chaque couple est complémentaire. Par exemple AACGCTT ne pourra se lier qu'à TTGCGAA.

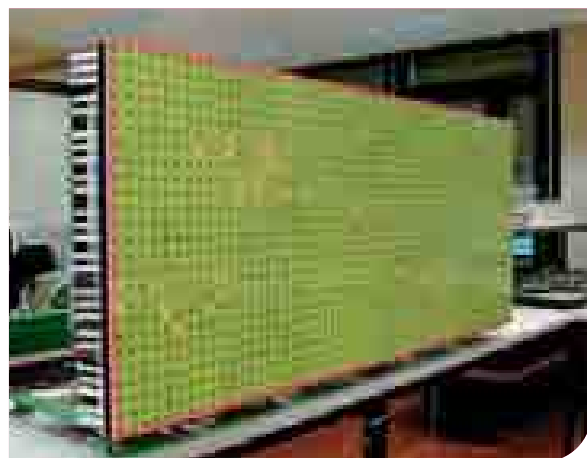
La différence majeure entre l'ordinateur classique et l'ordinateur biologique réside dans le traitement de

cette information. Tout d'abord il faut réaliser que le traitement chimique est infiniment plus lent que l'électronique. On parle d'heures, voire de jours ou de semaines. Mais le nombre de traitements parallèles peut atteindre plusieurs milliards..... Dans les 10 dernières années, deux stratégies successives ont vu le jour : la première (historiquement) a consisté à utiliser les molécules d'ADN pour encoder toutes les solutions possibles d'un problème d'une part et ses contraintes d'autre part, puis laisser les réactions chimiques éliminer les solutions ne répondant pas aux contraintes. Le plus fameux des problèmes ainsi traité est celui dit du voyageur de commerce (optimisation de la tournée d'un VRP qui doit visiter plusieurs villes, sans repasser deux fois par la même, et en parcourant la distance la plus courte).

La seconde stratégie a consisté à recréer les portes logiques de base auxquelles nous sommes habitués. La différence majeure avec nos portes classiques électroniques vient du fait que ces portes ne peuvent être réinitialisées une fois utilisées. Néanmoins, pour la plupart des problèmes, cela suffit.

Un bon ordinateur doit disposer d'organes d'entrées-sorties. Sans surprise, les entrées peuvent être constituées de brins d'ADN. Ceux-ci vont déclencher la séquence (chimique) de calcul.

Les sorties sont de plusieurs sortes,



Prototype du BioWall, avec 2000 molécules.

en fonction du type de résultat recherché. MAYA est un joueur de morpion. Les brins d'ADN contiennent à une extrémité une enzyme fluorescente, et à l'autre un inhibiteur. Quand le brin est coupé, suite à une opération logique, la fluorescence étant éloignée de son inhibiteur s'éclaire. Nous sommes là dans une utilisation informatique " classique ", mais est-ce la plus utile pour ce type d'architecture ? Surtout quand ce calculateur met environ 15 minutes par coup. Néanmoins, de nouvelles approches comme celle d'Erik Winfree de Caltech permettent d'améliorer les performances d'un ordre de grandeur. Il est amusant de noter qu'il se réfère pour sa méthodologie à une thèse de Claude Shannon de... 1940, époque où les portes logiques étaient réalisées avec des relais électriques. Quelle autre classe de problème pouvons-nous espérer résoudre avec ce type de technologie ? Imaginons que l'on code les brins d'entrée de notre circuit biologique en fonction de molécules présentes dans une cellule humaine. Après un traitement approprié, on pourrait imaginer de libérer une substance chimique, si les conditions sont remplies. Or les maladies se traduisent par la présence de différentes molécules bien identifiées à l'intérieur de nos cellules. On pourrait penser fabriquer des ordinateurs moléculaires injectés, capables de surveiller et de traiter les maladies au niveau le plus bas.

Et bien sûr, la science-fiction ne nous a pas attendus. Il existe même un ordinateur à base d'ADN qui sait faire le café : moi

■ Gilbert Vidal - C.T.O. idSM

La complexité simplifiée

Compilez et personnalisez pour le traitement haute performance.
Créez le logiciel le plus rapide possible et utilisez les dernières technologies dans les architectures Intel.

NOUVEAU | Intel® Compiler Suite 11 Professional Edition
pour Windows* et pour Linux. Contient les compilateurs Intel C++ et Visual Fortran, Math Kernel Library (Intel MKL), Intel Integrated Performance Primitives (Intel IPP) et Intel Threading Building Blocks (Intel TBB).

NOUVEAU | Intel® Cluster Toolkit Compiler Edition 3.2.
Fournit un package de logiciels contenant les compilateurs Intel C++ et Fortran pour toutes les architectures Intel. PLUS tous les outils Cluster Intel qui vous aident à développer, analyser et optimiser la performance de vos applications parallèles sous clusters Linux ou Windows Compute Cluster Server (CCS).

Intel® Math Kernel Library est une librairie de traitement numérique hautement optimisée pour les applications mathématiques, scientifiques, engineering et transactionnelles.

Intel® MPI Library fournit une implémentation flexible de MPI pour un développement multithreaded, providing interface plus élevée sur des architectures multi-algèbres.

WWW.SOSDEVELOPERS.COM/INTEL.HTM

Intel Compiler Suite Windows ou Linux + maintenance 1 an 733 € ht

Intel Cluster Toolkit Compiler Edition + maintenance 1 an 880 € ht

Intel Math Kernel Library Windows + maintenance 1 an 293 € ht

Intel MPI Library + maintenance 1 an 339 € ht



Dans une dizaine
d'années,

un programmeur
qui ne pensera pas
d'abord "portabilité"
ne sera pas
un programmeur.

James Reinders

Chief Software Evangelist

at Intel Software Products

COMSOFT direct

Bechtle's Software Specialist

**SOS
DEVELOPERS**

Plus d'informations sur :

www.sosdevelopers.com/intel.htm

Tel : 0825 07 06 07 • infos@comsoft.fr

TechDays 2009 : objectif .Net 4 et Windows 7



Les chiffres parlent d'eux mêmes : 16 000 visiteurs (dont 10 000 visiteurs uniques), 313 intervenants, 320 sessions, une centaine d'exposants, des plénières à 4000 personnes...

Microsoft n'avait pas de nouveaux produits à proposer mais plutôt une séance de rattrapage sur les outils existants, sur les tendances des prochains mois et le futur vu par l'éditeur. Windows Server 2008, SQL Server 2008, .Net 3.5 SP1, Visual Studio 2008, Silverlight 2 furent ainsi mis à l'honneur dans de nombreuses sessions. Durant ces trois jours, l'éditeur a aussi rappelé un élément majeur de sa stratégie : le cloud computing, le Software + Services. Si aujourd'hui, cela paraît encore un peu abstrait pour les utilisateurs et les développeurs, dans les prochains mois, les briques techniques et fonctionnelles seront disponibles, mais de l'aveu même de Microsoft, un gros travail de sensibilisation et d'évangélisation reste à faire sur Windows Azure, Mesh et les nouveaux services Live !

Des exposants actifs

L'éditeur **Netfective** a profité des Techdays pour expliquer et présenter **Blu Age** dans un contexte .Net 3.5 et notamment comment générer des applications RIA .Net avec l'outillage maison. Occasion pour rappeler l'approche MDA et son intérêt dans un projet informatique. **Intel** a pu dévoiler au public son futur outil de développement multicore : Intel Parallel Studio. **Soft-Fluent** a lui aussi mis à profit le salon pour présenter **CodeFluent 2009**, ce qui ne manqua pas de susciter un vif intérêt chez les visiteurs. L'une des attractions de la partie exposant fut le stand **Wygwam** avec la table Surface, une foule nombreuse s'est amassée pour voir et toucher cette technologie multitouch, finissant de convaincre de son intérêt et de son usage.

Cap sur Windows 7

Cela n'aura échappé à personne, Microsoft veut clore le chapitre Vista et les TechDays en furent une illustration, avec un grand nombre de sessions techniques et de présentations du prochain Windows. Le succès a été immense, jusqu'à 4 000 inscrits pour les sessions ! Les attentes sont nombreuses et les visiteurs ont pu voir les nouveautés (interface, fonctions), le multitouch a été largement mis en avant ainsi que la nouvelle interface et particulièrement le " Dock " Windows... Bien entendu, pour le développeur, Windows 7 a été aussi l'occasion de mieux appréhender les changements...

Côté développeur : l'embarras du choix !

Le développement a été fortement présent durant les 3 jours. Les sessions plénières et les autres ont largement rappelé l'importance de .Net 3.5 SP1 et l'intérêt de s'y mettre rapidement car il sera par défaut dans Windows 7 et apporte de nombreuses nouveautés. L'éditeur a aussi tenté de convaincre les développeurs de passer aux interfaces WPF et non plus Winform, la transition étant plus lente que prévue. Mais le fait est que WPF, le moteur graphique de l'éditeur Visual Studio 2010 devrait aider le mouvement.

Objectifs : Visual Studio 2010, F# et .Net 4.0

.Net 4 constituera réellement une mini révolution pour les développeurs avec l'introduction de la CLR 4.0, première CLR avec la v2 !

Un des points à retenir est l'important travail réalisé sur l'interopérabilité entre les différentes versions de la CLR. Concrètement, cela signifie que l'on pourra embarquer dans un unique process les différentes CLR (et donc les faire cohabiter sans problème). L'autre nouveauté concerne l'introduction du langage fonctionnel F# ! Les sessions autour de ce langage furent très suivies même si pour beaucoup, F# reste obscur dans son approche et son modèle de développement. Car on change totalement de philosophie de développement mais les avantages de l'utiliser sont proportionnels à la difficulté d'apprentissage. Pour les VBistes, Gregory Renard de Wygwam les a rassurés en présentant les nouveautés dans VB10 et elles sont nombreuses ! Désormais, VB et C# évoluent de la même manière... Ouf ! Une autre curiosité fut la session consacrée à la modélisation et l'approche de Microsoft qui va inclure trois directions convergentes : DSL, UML et OSLO. L'arrivée de l'UML est attendue avec impatience même si la concurrence se prépare à une rude bataille et si le premier support UML sera encore limité. L'arrivée de Microsoft peut aussi stimuler le marché et généraliser UML auprès des développeurs. Mais tout le monde regarde déjà au-delà avec OSLO qui apparaît de plus en plus comme une couche de modélisation totalement transverse qui sera supportée par les environnements et outils Microsoft et non-Microsoft. Oslo apporte une vision globale d'un projet, d'une application, de son système d'information. Rendez-vous en 2010 !

■ François Tonic

“ Avec l'atelier BLU AGE de transformation de modèles UML2, produisez instantanément des applications métiers 100% conformes aux besoins de vos utilisateurs ! ”

Modèle = Application

Application = Modèle

Avec BLU AGE®, valorisez enfin vos connaissances en UML !
Rejoignez la révolution des développements agiles !

BLU AGE_®
AGILE MODEL TRANSFORMATION

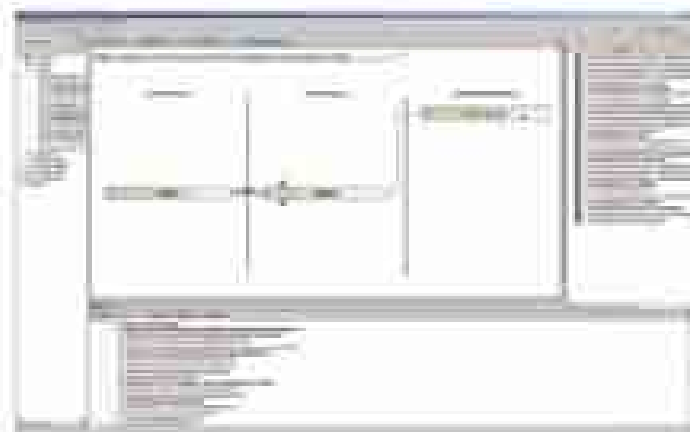
Informations et évaluation gratuite :

01 56 05 60 91

www.bluage.com

Éclipsez vos problèmes. Rationalisez vos développements !

- Génération de scripts du code source de vos applications.
- Génération vers la majorité des frameworks JAVA et .Net existants.
- Transformation et débogage de vos modèles UML2 en temps réel.
- Intégration native avec les IDE Eclipse et IBM Rational.
- Extraction du modèle FIM des applications existantes pour modernisation.
- Intégration avec les moteurs UML2 du marché.
- Génération de BPM.
- Nouveau ! Génère vos applications riches Microsoft .NET (RIA/NAM).



Construit sur Eclipse, BLU AGE® Edition 2009 transforme instantanément vos modèles UML en applications JAVA EE et .Net.

Conforme au MDA, BLU AGE® outil vos développements agiles et vos projets de modernisation d'applications.



BLU AGE est financé par l'Etat Français.
L'Etat : l'Agence de l'Innovation avec le Fonds National de Développement Régional

Windows 7

Il faut s'y préparer dès maintenant !

Si la transition entre Vista et 7 s'annonce plus rapide et plus simple qu'entre XP et Vista, ce n'est pas une raison pour le développeur de se désintéresser de cette nouvelle mutation qui se prépare d'ici 10 mois. Commençons dès aujourd'hui à nous y préparer, explorer les arcanes du système, les nouvelles API.

Souvent comparé à un Vista 2, Win7 est très attendu pour effacer l'accueil plus que mitigé de Vista en entreprise et chez les professionnels. Même s'il avait des qualités, Vista a souffert de son retard et surtout des problèmes de compatibilité de pilotes matériels et applicatifs. Car il ne faut pas oublier que Vista cassait le modèle applicatif présent dans XP. Les gourmandises du système, notamment pour les effets graphiques et certaines lenteurs font que le système traîne depuis son lancement une mauvaise réputation même si le service pack 1 améliore les choses. D'autre part, durant le développement, Vista avait perdu un nombre important de technologies et de nou-

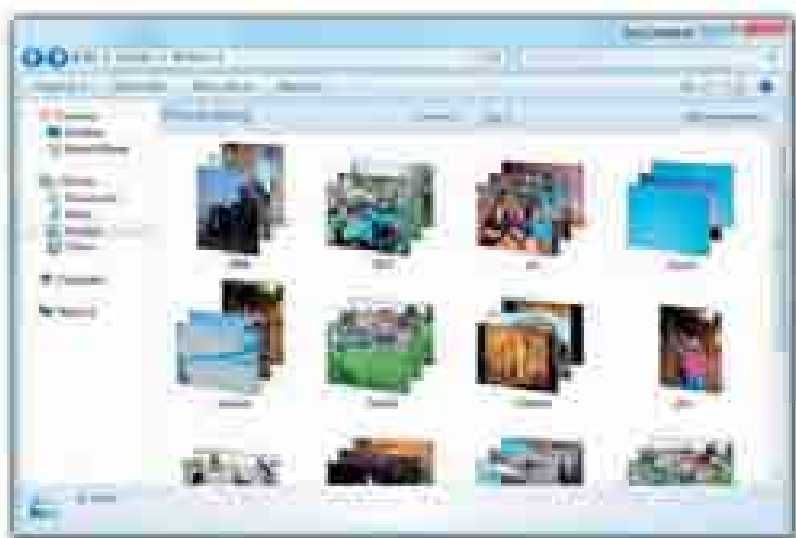


Fig.1



Fig.2



veautés promises dans le projet Longhorn... Win7 repose sur des fondations sans doute plus saines, même s'il faudrait attendre " Win8 " pour voir apparaître l'architecture MinWin qui doit rendre plus modulaire et plus légère la couche noyau... Par contre PowerShell 2.0 fait partie intégrante de Win7, tout comme IE8.

Compatibilité

Nous avons pu utiliser Windows 7 version PDC et la bêta. Pour les applications du marché nous avons jusqu'à présent eu peu de soucis d'installation ou d'utilisation. Par contre, nous avons rencontré des problèmes d'installation et de mise à jour sur des outils de développement en version française et des SDK ne voulaient pas reconnaître Win7 (ex. : Azure SDK).

Sur les formats de fichiers, notons :

- Open Packaging Conventions : Win7 supporte tous les formats (Microsoft et non Microsoft). Nouvelle API Packaging disponible.
- XPS : ce format concurrent de PDF est pleinement disponible pour les applications. Le développeur peut

générer du XPS depuis son application. API accessible en Win32 (nouveau) et .Net

Côté internet, les utilisateurs emploieront par défaut IE8.

Pour le développeur, IE8 constitue un certain fossé fonctionnel et de standard, même si le navigateur intègre plusieurs moteurs de rendu pour faciliter l'affichage des sites optimisés pour IE6 ou 7.

Le développeur web aura un travail de test, d'ajustement et de développement spécifique pour les fonctions (ex. : les webslice) propres à ce navigateur. D'autre part les applications fonctionneront le plus souvent normalement sur Win7 et Windows Server 2008 r2.

Notons le retrait du système de Windows Mail et de l'API CoStartOutlookExpress. Donc prudence sur les fonctions et API dépréciées !

Nous vous conseillons vivement de parcourir le Windows Application Quality Cookbook. Il liste tous les changements systèmes et la criticité de ceux-ci pour le développeur. Un exemple : quel impact a le remplacement du dossier Documents par le

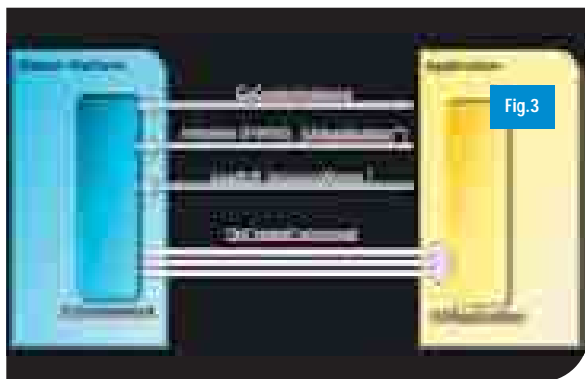


Fig.3

" File Library " ? Il faut utiliser l'espace de nom Shell, l'interface IShellLibrary...

De nouveaux réflexes

L'interface évolue avec Win7, tout particulièrement sur le bureau, avec une barre de menus qui se transforme en véritable Dock à la Mac OS X. Il faut donc tout d'abord connaître les nouveautés comme le Jump List, les barres de progression, les gadgets, etc. Ainsi le Jump List est un mini-menu démarrage dédié à une application. Il est personnalisable. On peut accéder aux documents récents de l'application. Deux fonctions sont disponibles : la destination (catégories de fichiers), les Tasks. [Fig.1]

En fait c'est toute l'Application Identity (ou Application ID) qui a été revue dans Win7 afin d'unifier l'ensemble. Chaque application doit avoir le même système de raccourcis claviers, la même présentation dans le Dock, etc. Dans la barre des tâches Windows, on peut aussi avoir une mini-fenêtre de visualisation (ex. : une photo, une vidéo). Cette fonction (Thumbnail Toolbars) est accessible via une API.

Une autre nouveauté, bienvenue, est la possibilité d'indiquer sur les icônes d'une application une information, une notification (c'est l'overlay Icons).

Toujours sur les icônes, on dispose d'une barre de progression directement sur l'icône d'une application ! Pratique à l'usage. Autre grosse nouveauté dans la barre des tâches, la possibilité de switcher entre les fenêtres ouvertes dans une application. Ces nouvelles fonctions sur les icônes obligent à posséder des icônes de bonne qualité.

D'autre part, le bureau Win7 supportera : WPF 4.0, MFC 7 et bien entendu Windows 7 Platform SDK. Microsoft conseille aux développeurs de regarder les exemples de code disponibles dans le nouveau SDK (fichier ISO à installer sur Win7).

Un ruban Windows ! [Fig.2]

Autre nouveauté importante au niveau interface : le Windows Scenic Ribbon. Il s'agit de reprendre le principe du ruban introduit avec Office

2007. Cette " barre d'outils assez volumineuse " permet d'accéder d'un clic aux principales fonctions du logiciel grâce à des boutons, onglets, listes déroulantes, etc. Ce ruban Windows devient un standard sous Win7 qu'il sera aussi possible d'utiliser sous Vista avec son application. Ce ruban utilise Win32 et COM mais MFC et WPF sont aussi supportés ! Les développeurs natifs sont les premiers visés (C, C++). Attention à bien respecter les bonnes pratiques ergonomiques et fonctionnelles pour ne pas faire n'importe quoi ! Pour aider le développeur, un modèle MVC est implémenté pour séparer les différentes couches. En gros : on a le Windows Scenic Ribbon l'API COM (initialisation et événements), le balisage (pour l'organisation des contrôles) et dans son code la logique applicative. [Fig.3]

L'accessibilité a aussi été revue avec de nombreuses améliorations afin de faciliter l'utilisation du système et des applications par des personnes ayant un handicap. On dispose d'un outil de vérification pour s'assurer que son application est conforme aux règles d'accessibilité (UI Accessibility Checker). [Fig.4]

Des web services natifs

Autre changement notable dans Win7 : la possibilité de créer des services web en code natif, donc en Win32, sans aucune dépendance à .Net ! On garde l'accès aux piles SOAP et WS-* (série de spécifications sur la sécurité, l'accès, etc.), à WCF, aux services web ASP.Net XML. Cette API Windows Web Services fonctionnera aussi sur XP, Vista, 2003 et 2008.

Cette API propose trois couches :

- Service Model : pour gérer automatiquement la communication (service proxy du client et le service host sur le service lui-même), échanges de messages comme s'il s'agissait d'appels de fonctions.
- Channel Layer : contrôler la communication, les messages et donc les canaux avec abstraction du protocole d'échange des messages, API unifiée pour le transport (http, tcp, udp).
- Couche XML : accès complet au contenu des messages, API unifiée pour l'encodage, pas de support du DTD, présence du xmlbuffer, du xmlreader et du xmlwriter.

A noter deux points importants :

- la sécurité passe par un cryptage du transport (SSL et Windows SSPI) mais aussi une sécurité au niveau du message même. Par contre, pour le moment il n'y a pas de mode de sécurité complet sur les messages. Ce point sera à surveiller.
- Sérialisation : mapping entre les types de données en C vers ou provenant des types XML. Bon à savoir : génération des types C à partir d'un XSD.

Une instrumentation améliorée

Win7 arrive avec de belles nouveautés sur les outils d'instrumentation : nouvelles API 2.0 du Kernel Mode, amélioration des performances et robustesse, Scripting avec PowerShell. Le développeur pourra aussi mieux cerner les problèmes et dysfonctionnements de son application grâce à une implémentation étendue, via le *Troubleshooting Pack*. Ce pack peut être lancé soit par l'utilisateur, soit par l'application. Il détecte et résout les problèmes de configuration. Et vérifie que la configuration est propre (= conforme aux besoins, et sans bizarrerie). D'autre part, il détecte sur les problèmes logiciels : les problèmes de conception et les conflits de configuration. La compatibilité avec PowerShell permettra de détecter, résoudre et vérifier le code, de capturer les entrées de l'utilisateur. Il est possible de déployer avec son application le *Troubleshooting Pack*. Ces évolutions devraient faciliter la vie du développeur !

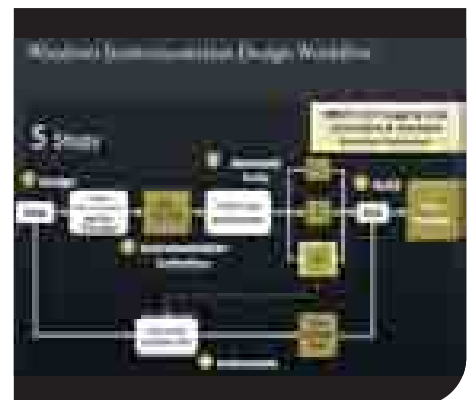




Fig.4

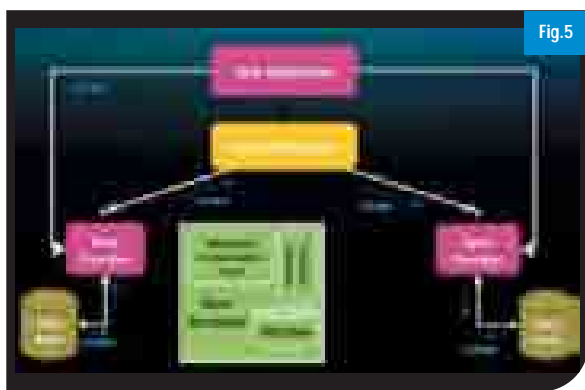


Fig.5

Windows 7 bêta : un succès

Le téléchargement libre de la version bêta de Windows 7 est désormais clos depuis le 10 février, mais reste possible via MSDN et TechNet. Microsoft annonce plusieurs millions de téléchargements (et de DVD) de cette pré-version, cependant l'éditeur ne fournit aucun chiffre précis. En France, tout juste savons-nous que notre pays est le 3e plus actif sur la bêta avec plusieurs dizaines de milliers de téléchargements, derrière les Etats Unis et l'Allemagne. Le chiffre de 100 000 bêta en France a été prudemment avancé, alors qu'aucune version française n'était disponible.

Pour Microsoft, cette bêta constitue un test grandeur nature pour l'interface, les applications, les pilotes, qui permet d'avoir des commentaires et remontées des éditeurs, développeurs et utilisateurs. Visiblement, la bêta plaît pour sa rapidité et sa stabilité. Microsoft ne fournit aucun chiffre ou détail sur les commentaires remontés et encore moins sur les bugs découverts.

Sur la disponibilité de Win7, personne ne s'avance réellement, mais tout indique que la version RTM (version prête à être dupliquée) sera disponible avant les fêtes de fin d'année, peut être dès octobre ou novembre. Soit 3 ans après Vista comme le promettait Steve Ballmer. Quant à la disponibilité générale (donc pour tout monde), rien n'est précisé, peut être avant la fin décembre ou début 2010.

Le déploiement

Déployer une application sur Win7 ressemble à l'installation sur Vista. On notera l'arrivée de MSI 5.0. Le processus d'installation a été simplifié pour supprimer la complexité introduite dans Vista. MSI 5 reprend les mécanismes de MSI 4.5 et de .net 3.5 SP1 avec des performances annoncées à la hausse ! Surtout le fonctionnement UAC a été quelque peu amélioré.

Autre point intéressant : le Per-User Application. Cette technique doit simplifier l'installation et la désinstallation par l'utilisation. Les applications s'installent dans le profil utilisateur, sans impacter un autre compte utilisateur. Le Per-User signifie que l'on installe une application pour un utilisateur et non l'ensemble des utilisateurs (ce que l'on fait avec le Per-Machine). MSI 5 supporte cette technique d'installation. Le Per-User intervient dans l'installation d'un service ou d'un pilote. Théoriquement, on améliore la sécurité (on n'exige pas de privilège élevé) et on évite UAC (dixit Microsoft).

Un UAC amélioré

Le User Access Control (UAC) fut une des nouveautés les plus critiquées par les utilisateurs. Win7 améliore les choses et il devient un peu moins intrusif à l'usage. Il doit (notamment) permettre d'éviter d'installer des applications avec un niveau de privilèges élevés, de type administrateur, provoquant un risque de sécurité. Car un utilisateur " normal " n'a pas besoin de privilèges élevés. Il s'agit donc pour le développeur de coder en " utilisateur standard ". Aux applications de ne pas exiger des privilèges inutilement élevés...

Synchronisation avec les terminaux

L'un des objectifs de Win7 est de faciliter les échanges des données entre les différents terminaux. Pour cela, il faut disposer d'une plate-forme la plus unifiée possible, une couche technique que les éditeurs et constructeurs peuvent intégrer. Et créer une nouvelle expérience utilisateur. Cela passe par l'usage, par défaut, du Framework Sync. En réalité, Win7 inclut plusieurs composants, en dehors de Sync : Windows Portable Devices (WPD), Media Transfer Protocol et un Data type Schema. WPD est une technologie de pilote destinée aux terminaux portables (au sens large). Pour rappel, Sync permet de se passer d'un serveur, on peut réaliser une synchronisation terminal à terminal. Win7 possède deux interfaces Sync : ISyncKnowledge et ISyncProvider.

Vivement que Mesh devienne un standard d'échange et de synchronisation !
[Fig.5]

Win7 et le multitouch

Impossible de ne pas évoquer la technologie multitouch dans le nouveau système de Microsoft. Bien entendu il faut que le développeur et l'utilisateur disposent d'un écran multitouch. Créer une telle application nécessite de penser l'ergonomie et les actions multitouch que l'on souhaite inclure dans son application. Win7 expose des API Multitouch ainsi que dans .Net 4 et dans le Surface SDK. D'ailleurs, en développant avec .Net 4, son application " touch " sera exécutable sur Surface et Win7. Bien entendu il est possible de passer en Win32, Winforms. Il y a aussi différents niveaux d'implémentation :

- une bonne : on intègre les contrôles standard, peu de choses à modifier
- une meilleure : on intègre les gestes (ou *gesture* en anglais).

- Mais l'ultime intégration est d'adapter son application en implémentant notamment les événements touch.

Son application devra intégrer la gesture du multitouch c'est-à-dire les gestes que l'utilisateur peut faire. .Net 4 proposera de nouveaux contrôles spécifiques. Bref un travail d'apprentissage sera nécessaire !

Dans les prochains numéros nous aurons l'occasion de revenir plus en détail sur Win7 et les nouvelles API !

■ François Tonic



FarPoint Spread for Windows Forms | à partir de € 710

FarPoint

Feuille de calcul complète pour applications Windows Forms.

- Contrôle unique, 2 milliards de feuilles, avec chacune 2 milliards de lignes et 2 milliards de colonnes
- Renseignement automatique : anticipation de la frappe dans la cellule
- Nouveaux : export PDF, groupements, liste de formules, Excel 2007 XML
- Inclut une version pour .NET 2.0 et .NET 3.5 (Visual Studio 2008)



Nevron Chart for .NET | à partir de € 452

NEVRON

Puissante fonctionnalité graphique dans vos applications et pages Web.

- Graphiques 2D/3D : histogramme, courbe, palier, surfacique, secteurs, points, bulles, stock, barre flottante, radar, poterie, min/max, maillage, etc.
- Support conceptuel VS pour modifications rapides via interface visuelle
- Associez les types de graphiques 2D et 3D dans un contrôleur graphique
- Fournissez fonctionnalités avancées d'édition comme assistants et éditeurs de style



DXperience Enterprise | à partir de € 997

DevExpress

Tous les outils DevExpress ASP.NET, WinForms et IDE Productivity en un.

- Abonnement de 12 mois pour tous les produits et mises à jour Developer Express et accès aux versions bêta en développement actif
- Composants et outils : grilles, entrée de données, outils d'écriture de code, analyse de données, graphiques, navigation/disposition, planification, solutions reporting, bibliothèques d'impression, outils de rapprochement, bibliothèques ORM



TX Text Control .NET and .NET Server | à partir de € 413

TX TEXT CONTROL

Le traitement de texte pour Visual Studio .NET.

- Le traitement de texte professionnel pour vos applications
- Zones de texte Windows Forms hors droits
- WYSIWYG, tableaux imbriqués, cadres, en-têtes, pieds de pages, images, notes, listes numérotées, num, sauts de section, etc.
- Opérations aux formats DOCX, DOC, RTF, HTML, TEXT et XML

© 2009 ComponentSource. Tous droits réservés. Toute la documentation ou autres logiciels sont la propriété exclusive de ComponentSource ou de l'un de ses fournisseurs.

Séjour exclusif en France
ComponentSource
30 Grenville Road
Reading,
Berkshire
RG2 1PE
Royaume-Uni

Séjour exclusif aux États-Unis
ComponentSource
850 Clematis Park Way
Suite 100
Woodstock
GA 30189-4001
États-Unis

Séjour exclusif au Japon
ComponentSource
3F Higashinaka Bldg
3-3 Higashinaka Chome-4
Tama
Tokyo
164-0861

Numero vert:

0800 90 92 62

www.componentsource.com

ComponentSource est membre de

Partnership for Software Development

Partnership for Software Development

Partnership for Software Development

Partnership for Software Development

Base de données

- ✓ optimiser
- ✓ sécuriser
- ✓ relier



© istockphoto.com/Nishan Sothlingam

Aujourd'hui, la base de données est plus que jamais au cœur des applications. La donnée, plus que le SGBD, est devenue un enjeu central. Comment traiter une masse d'informations toujours plus grande ? Mais au-delà de la simple volumétrie des données, de nombreuses problématiques se font jour : sécurité, intégrité, optimisation des requêtes, amélioration des performances, optimisation du stockage, mapping objet / relationnel.

La notion de temps réel est illusoire

De nombreuses évolutions sont possibles et transforment déjà le monde des SGBD. Les performances et l'accès immédiat aux données constituent deux écueils importants. La notion de temps réel est illusoire, et ce à plusieurs titres : qu'appelle-t-on temps réel ? Quel temps de latence accepte-t-on ? Que faire des contraintes physiques du matériel ? N'oubliez pas que l'on ne peut pas théoriquement passer outre les Entrées / Sorties ou encore la lecture / écriture des données sur un disque dur. Nous disons bien théoriquement, car différentes techniques permettent de contourner partiellement ces contraintes : le cache ou encore la mémoire. Ce dernier cas de figure nous paraît particulièrement intéressant pour accélérer l'accès aux données. Ce n'est pas un hasard si la notion de

SGBDM ou base de données mémoire (ou en mémoire) refait surface. Aujourd'hui, les solutions hybrides SGBDR + fonctions en mémoire fleurissent. Courant 2008, IBM avait racheté SolidDB, un acteur reconnu des SGBDM.

Aujourd'hui, les données géospatiales tiennent une place de plus en plus grande, même si cela implique parfois une explosion de la volumétrie, obligeant à toujours mieux gérer les données de type Blob. La volumétrie pose d'ailleurs de réels problèmes aux développeurs et DBA : comment stocker une telle quantité de données sans exploser les tables et les bases ? A cela les éditeurs offrent deux réponses : la compression et le partitionnement. Nous verrons notamment dans notre dossier en quoi le partitionnement, sous MySQL, constitue une excellente réponse pour répartir les données d'une même base, d'une même table entre plusieurs serveurs... La compression est une autre fonction particulièrement choyée par les SGBD. Nous aborderons aussi l'éternel problème du mapping relationnel objet avec un véritable guide de survie du développeur sur Hibernate. Qui a dit que le mapping était un jeu d'enfant ?

■ François Tonic

Dans les prochains numéros nous aborderons : Hibernate, le futur des SGBD et comment utiliser Linq avec une base MySQL !



Les mécanismes de sécurité dans les bases de données

Le monde de la finance a toujours été un moteur pour faire évoluer la gestion et la sécurité des données. L'incendie de la salle du marché du Crédit Lyonnais en 1996 avait ainsi ouvert la voie à une révision complète des plans de secours et de reprise d'activité dans l'ensemble des services informatiques. Ce sont désormais les thèmes autour de la sécurité (accès aux données, surveillance des utilisateurs...), qui sont au cœur des préoccupations depuis les dernières affaires, notamment aux Etats Unis. Pour cela un certain nombre de directives ont vu le jour (Sarbanes Oxley, HIPAA, ...). La base de données est au cœur de cette problématique de par son rôle de stockage et de diffusion des informations critiques de l'entreprise.

Cet article a pour but de faire un état de l'art des fonctionnalités associées à la sécurité, disponibles sur les différents SGBD du marché (Oracle, SqlServer, DB2, Sybase et MySQL). Le schéma [Fig.1] symbolise les différents éléments de sécurité pouvant être mis en œuvre autour de la base de données.

Authentification

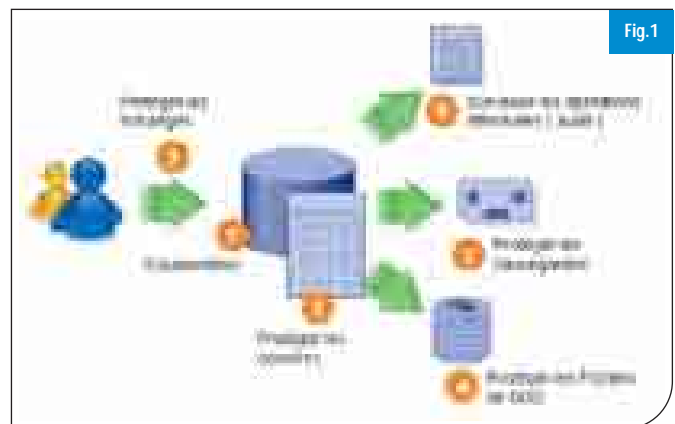
L'authentification est l'opération qui consiste à vérifier la légitimité d'une demande de connexion par un utilisateur. En général tous les SGBD permettent de déclarer les utilisateurs directement au niveau de la base à travers une notion de user, login et mot de passe. Pour certains SGBD, l'authentification se fera directement avec le compte système. Gérer les comptes utilisateurs devient complexe quand le nombre de bases et d'applications devient important. C'est pour cela que les éditeurs proposent des solutions permettant de s'interfacer à des sources extérieures de type annuaire LDAP (), Active Directory, etc. qui vont centraliser la gestion des utilisateurs. Cela permet d'en simplifier la gestion.

Le principe d'utilisation d'une source externe est sensiblement la même pour les éditeurs de SGBD, en général il faut installer un module spécifique qui très souvent demande une licence supplémentaire afin d'établir le dialogue avec l'annuaire.

Une gestion centralisée des utilisateurs permet également de mettre en place une solution d'identification unique ou SSO (Single Sign On) qui permet à un utilisateur une fois identifié d'utiliser différents services ou applications.

Le schéma [Fig.2] décrit un exemple d'architecture qui met en relation une base de données, un annuaire de type active directory et un serveur d'authentification de type Kerberos. Le schéma [Fig.2] décrit la démarche d'authentification dans ce type d'architecture.

- Les utilisateurs de la base de données sont assignés à l'Active Directory
- Le mapping des utilisateurs permet à un système d'exploitation d'accéder à une base en tant qu'utilisateur de l'Active Directory
- L'authentification va se faire avec Kerberos
- Avec Kerberos, un client Windows s'authentifie sur l'Active



Directory via le serveur Kerberos, le kdc (Microsoft Key distributor) puis il reçoit un ticket Kerberos.

- Si l'utilisateur veut se connecter au serveur de BDD, celui-ci vérifie auprès du serveur Kerberos si l'utilisateur est déjà connecté. Si c'est déjà le cas, le serveur de Base de données récupère le ticket Kerberos et l'utilisateur peut se connecter sans spécifier de mots de passe.

La difficulté de la mise en place d'une architecture SSO, réside plus sur la partie cliente que la partie serveur base de données. En effet pour qu'une application puisse fonctionner en SSO, il faut le prévoir lors du développement.

Protection des échanges

La protection des échanges consiste à sécuriser les informations échangées entre l'utilisateur et la base de données. Le principal protocole de transport pour l'échange de données entre un client et un SGBD sur système ouvert est TCP/IP. La plupart des solutions de cryptage sont transparentes pour les applications. SSL (Secure Socket Layers) et sa nouvelle dénomination TLS, est le protocole de sécurisation le plus connu, notamment pour les échanges sur internet. Cette solution qui s'appuie sur l'utilisation d'un certificat numérique peut être également mise en œuvre pour les échanges avec la base de données. Tous les éditeurs de

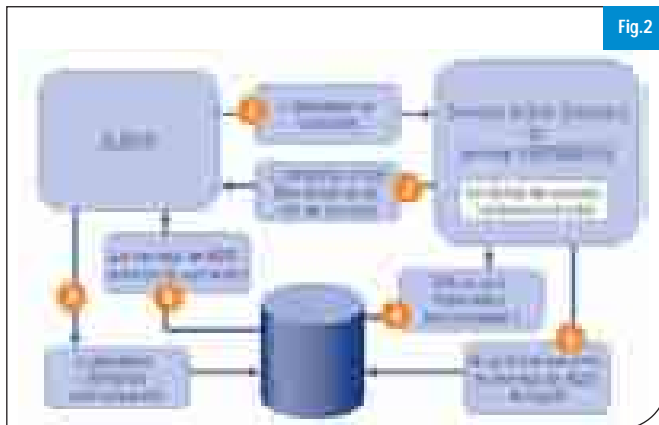


Fig.2

SGBD de notre étude, permettent la mise en œuvre directe ou à travers un module complémentaire d'une sécurisation SSL.

Protection des données

La sécurisation des données permet de crypter ou de limiter la vision des données pour un utilisateur au niveau d'une base, d'une table, d'une colonne. La norme SQL ANSI permet déjà à travers la gestion des droits (GRANT, REVOKE) d'autoriser un utilisateur à voir ou modifier des informations sur une table, ou sur certaines colonnes. Par contre les données sont stockées en clair, non cryptées. Dans le cas de données sensibles, il est important de proposer des solutions de cryptage des données. Pour répondre à ces problématiques, les éditeurs de SGBD proposent essentiellement deux solutions :

- La première consiste à utiliser des fonctions de cryptage directement dans le code SQL, avec suivant les algorithmes employés, l'utilisation d'une clé. L'emploi des fonctions de cryptage oblige à une modification du code applicatif. Les principaux algorithmes disponibles suivant les SGBD sont DES, 3DES, AES, MD5, MD4, SHA et SHA-1.
- La deuxième consiste à mettre en place un système de cryptage connu sous le nom générique de TDE (Transparent Data Encryption). Cette solution de cryptage est transparente pour les applications (Il n'y a pas besoin de modifier le code applicatif). En général, cette solution permet de protéger les fichiers de la base ainsi que les sauvegardes.

SQL Server 2005 à travers ses fonctions de cryptage propose trois modes de fonctionnement basés sur des commandes SQL et des fonctions spécifiques. Pour mieux comprendre l'utilisation des fonctions de cryptage, nous allons détailler les différentes variantes disponibles :

- Utilisation d'un certificat :

```
CREATE CERTIFICATE cert_1 WITH ...
UPDATE ... SET colonne_cryptée = EncryptByCert(Cert_ID('
cert_1'), 'texte à crypter')...
SELECT ...DecryptByCert(Cert_ID('cert_1'), colonne_cryptée) ...
FROM ...
```

- Utilisation d'une clé

```
CREATE ASYMMETRIC KEY key_1 WITH ALGORITHM=RSA_2048 ...
UPDATE ... SET colonne_cryptée = EncryptByAsymKey(AsymKey_ID
```

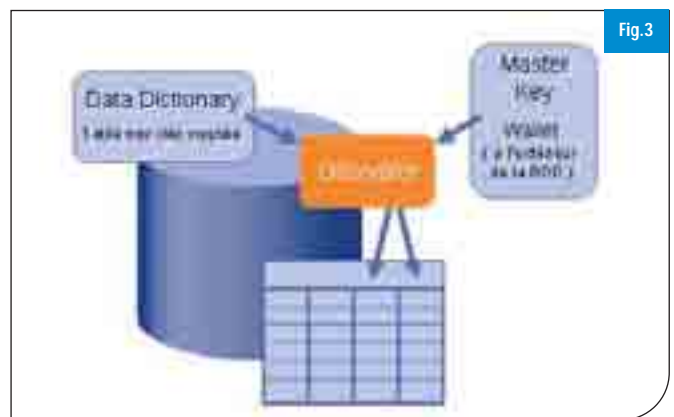


Fig.3

```
('key_1'), 'texte à crypter') ...
SELECT ... DecryptByAsymKey(AsymKey_ID('key_1'), colonne_cryptée)
... FROM ...
```

- Utilisation d'une clé et d'un certificat.

```
CREATE SYMMETRIC KEY key_2 WITH ALGORITHM AES_256
ENCRYPTION BY CERTIFICATE cert_1

OPEN SYMMETRIC KEY key_2 DECRYPTION BY CERTIFICATE cert_1
UPDATE ... SET colonne_cryptée = EncryptByKey(Key_GUID('key_
2'),'texte à crypter');

SELECT ... DecryptByKey(colonne_cryptée) ... FROM ...

CLOSE SYMMETRIC KEY key_2
```

En ce qui concerne TDE (Transparent Data Encryptions), actuellement seul Oracle, SQL Server et Sybase proposent ce mode de cryptage. Sous Oracle la fonctionnalité **TDE** qui fait partie de l'option **Oracle Advanced Security**, permet d'effectuer un cryptage au niveau de la colonne ou du *tablespace*. [Fig.3]

Pour cela, Oracle utilise une clé externe (master key) qui peut être stockée dans un Oracle wallet (protégé par mot de passe). Si un utilisateur accède à la base, les données seront décryptées automatiquement.

```
// Création du répertoire devant contenir le 'Wallet'

ALTER SYSTEM SET ENCRYPTION KEY AUTHENTICATED BY "MotDePasse";

CREATE TABLE test (
  id NUMBER,
  num_cb VARCHAR2(12) ENCRYPT
);

// Au démarrage et à l'arrêt de l'instance on ouvre et on ferme
l'accès au 'Wallet'. La fermeture du Wallet interdit la lecture
des données cryptées.

ALTER SYSTEM SET WALLET OPEN IDENTIFIED BY "MotDePasse";
```

ihm

EN TOUTE SIMPLICITÉ !

Pour construire des applications riches, accéder à des données hétérogènes (applications métier, SD, et bases de V4.1, Hibernatis, GED et sites documentaires), déployées en Swing, Java EE ou Oxyd, Apex et à l'extrême, s'adressant dans les processus d'entreprise.

Adoptez la puissance et l'agilité de l'approche Model-Driven

**NOUVELLE
VERSION
LEONARDI
V4.2
open source**

Construire avec la même agilité et la même puissance d'échange complexes dynamiques, modélisation du type de vie du logiciel, documents éditables par prototype pour faciliter les besoins, abstraction technologique, abstraction, après la réalisation, le tout avec une même technique simple.





```
ALTER SYSTEM SET WALLET CLOSE;
```

Sous Sybase la fonction TDE se fera au niveau de la table ou de la colonne, l'utilisateur n'aura accès aux données que si le DBO ou le SSO lui a attribué un droit spécifique.

```
CREATE ENCRYPTION KEY key_1 AS DEFAULT FOR AES

ALTER TABLE achat MODIFY num_cb NULL ENCRYPT WITH key_1

GRANT DECRYPT ON achat TO user2
```

La fonction TDE disponible sous SQL Server 2008 permet de crypter une base complète. Les données seront décryptées automatiquement aux utilisateurs qui se connectent à la base, il n'y a pas besoin de droits spécifiques.

```
USE master
GO
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'MotDePasse'
go
CREATE CERTIFICATE Cert_1 WITH SUBJECT = 'Mon certificat'
go
USE MaBase
GO
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_128
ENCRYPTION BY SERVER CERTIFICATE Cert_1
GO
ALTER DATABASE MaBase
SET ENCRYPTION ON
GO
```

Protection des fichiers

La protection des fichiers de la base de données, évite qu'une copie de fichier puisse être utilisée par un autre serveur de base de données. Cette solution consiste souvent en un cryptage complet à travers une solution de cryptage de type TDE. Certains éditeurs n'ayant pas de solutions réelles, proposent d'utiliser le cryptage assuré par le système d'exploitation, au niveau des File System. Cette solution est malgré tout souvent déconseillée pour des problèmes de performances.

Sécurisation des fichiers de sauvegardes

La sécurisation des fichiers de sauvegardes évite de pouvoir recharger facilement ces fichiers sur une nouvelle instance. En général une protection par mot de passe est proposée ou si une solution de cryptage complet de la base a été mise en place (TDE), la sauvegarde sera elle-même cryptée.

Audit

La surveillance de l'utilisation de la base consiste à consigner toutes les opérations qui sont effectuées sur la base. Des directives de type Sarbanes Oxley conseillent de consigner les opérations effectuées au niveau des systèmes d'information et notamment au niveau des bases de données. Des mécanismes d'audit sont implémentés dans tous les SGBD afin de tracer l'origine, l'heure et le but de chaque action, permettant ainsi de s'assu-

rer de la responsabilité des actions utilisateurs. En général, un paramétrage permet de filtrer les informations recherchées avant de les stocker dans une table. Suivant les solutions implémentées et le paramétrage choisi, l'impact sur les performances sera plus ou moins important.

- Oracle audite l'activité de la base de données par ordre SQL, par privilège système de l'utilisateur, par objet, ou par utilisateur. L'enregistrement de l'audit peut être stocké dans une table de la base de données. Pour les responsables DBA, Oracle peut auditer toutes les opérations SYS sur le système d'opération, afin de réaliser une revue complète de la sécurité administrateur et son audit. Il est également possible d'utiliser un audit plus fin avec l'option " **Fine-Grained Auditing** ", ce qui permet de générer un événement d'audit lorsqu'une table est par exemple accédée durant une tranche horaire ou sur une colonne spécifique.

Avec **Audit Vault**, Oracle automatise la collection d'informations d'audit et le processus d'analyse des informations consolidées. En plus des bases Oracle, cet outil supporte Microsoft SQL Server 2000 and 2005. IBM DB2 UDB 8.2 ,9.5, Sybase ASE 12.5 and 15.2 .

- SQL Server 2005 supporte trois types d'audit, le premier est basé sur la connexion, le deuxième, sur des triggers de modification de structure (DDL triggers), et le dernier sur des événements. L'audit basé sur les événements a un fonctionnement complètement asynchrone (utilisation du bus de messages XML Service Broker), ce qui lui permet d'avoir un impact très faible sur les traitements, contrairement à l'audit basé sur les triggers. La version 2008 apporte des mécanismes supplémentaires à l'audit, notamment une historisation dans de nouvelles tables systèmes (Change Data Capture, Change Tracking).
- Sybase propose un système d'audit performant, qui permet de consigner les opérations effectuées sur la base suivant le paramétrage qui a été mis en œuvre. Les informations sont stockées dans une série de tables (SYSAUDITS_01, SYSAUDITS_02 ...), ce qui évite la perte d'information d'auditing et facilite l'archivage.
- DB2 fournit un service d'**audit** qui permet d'enregistrer les opérations réalisées par les applications, les accès utilisateurs individuels et les administrateurs de base de données. L'administrateur de la base utilise l'outil db2audit pour configurer l'audit au niveau d'une instance ou au niveau de chacune des bases de l'instance. Cet outil permet également d'archiver les fichiers d'audit ou bien d'extraire des informations des fichiers archivés.
- Sous MySQL, il n'y a pas à ce jour de solution réellement aboutie. Le log général permet de tracer l'activité des utilisateurs, dans un fichier ou dans une table depuis la version 5.1, mais au détriment des performances. L'outil MySQL Proxy, pour l'instant en version Alpha, propose de se glisser entre le serveur et les clients afin d'intervenir à chaque niveau de communication. Il est alors possible, entre autres, d'auditer tout ce qui passe entre le serveur et les clients.

Limiter les droits des administrateurs (DBA)

On considère qu'actuellement les administrateurs de base de données ont trop de pouvoir sur la base de données. Les éditeurs de base de données ont développé de nouvelles fonctionnalités qui permettent de réduire la vision des DBA.

Sybase ASE avec sa version 15, apporte cette nouvelle fonctionnalité, à travers la notion de 'key custodian' qui permet à des utili-



sateurs de privatiser et de partager des données sans que l'administrateur puisse y accéder. Pour cela, un système de mot de passe et de déclaration permet d'échanger des clés entre divers individus.

```
// user1
CREATE ENCRYPTION KEY key_1 WITH PASSWD 'MotDePasse'
GRANT SELECT ON KEY key_1 TO dbo

// DBO
CREATE TABLE test (
  id NUMBER,
  num_cb VARCHAR2(12) ENCRYPT WITH user1.key_1 )

GRANT SELECT ON test TO user2

// user1
GRANT DESCRIPT ON test (num_cb) TO user2
ALTER ENCRYPTION KEY key_1 WITH PASSWD 'MotDePasse'
ADD ENCRYPTION WITH PASSWD 'user2_ps' FOR USER user2

//user2
SET ENCRYPTION PASSWD 'user2_ps' FOR KEY user1.key_1
SELECT * FROM test ...
```

Oracle avec son module Oracle Database Vault permet également de restreindre les privilèges des administrateurs (DBA) et de vérifier la conformité à certaines réglementations comme SOX (Sarbanes Oxley)

Informations complémentaires

Oracle, avec sa base de données Privée Virtuelle (**Virtual Private Database : VPD**) permet, au sein même de la base de données, d'avoir l'assurance de la séparation des données pour les accès aux données par utilisateur ou par groupe d'utilisateurs.

En ajoutant dynamiquement le prédicat aux ordres SQL (derrière la clause WHERE), la VPD limite les accès aux données au niveau ligne, et elle attache la politique de sécurité à la table (à la vue, ou au synonyme) elle-même. D'autres options et outils complémentaires sont également disponibles dans l'offre Oracle :

- **Oracle Database Vault** : Permet de restreindre les privilèges des super utilisateurs (dba..) et de vérifier la conformité à certaines réglementations ex SOX (Sarbanes Oxley)
- **Oracle Label Security** : Permet de mettre un niveau de sécurité à chaque ligne d'une table.
- **Data masking** : Permet de masquer les données de production (duplication de base, export)

La plupart de ces modules d'Oracle supposent d'être en version entreprise et demandent une licence supplémentaire :

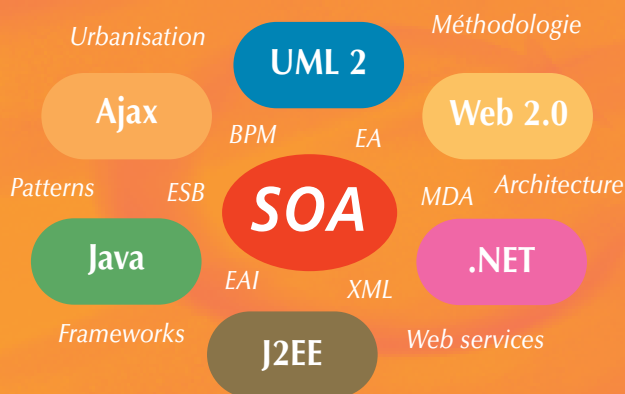
La plupart des fonctionnalités de sécurités associées à Sybase ASE sont présentes dans 2 modules supplémentaires :

- 'Security & Dir services' : permet de gérer l'authentification externe (Kerberos, PAM), ainsi que le cryptage SSL des connexions.
- 'Sybase ASE Encryption' : permet de gérer les différentes solutions de cryptage des données.

■ Yves MOULIN

Directeur Technique - Cap DATA Consulting

Soyez prêts pour les nouvelles architectures SOA et Web 2.0 !



SOFTEAM Formation

Calendrier complet et inscriptions en ligne

www.softeam.fr



Tél. : 01 53 96 84 30 - Fax : 01 53 96 84 01

Paris : 21 avenue Victor Hugo - 75016

Rennes - Nantes - Sophia Antipolis

SOFTEAM
Think Object

SOA est devenu en peu de temps le mot-clé des développements logiciels. SOA est une nouvelle façon de faire qui s'appuie sur un ensemble de technologies existantes : UML, J2EE, .Net, XML, etc. Maîtriser SOA implique de maîtriser ces technologies pour les associer efficacement au sein d'une nouvelle approche.

SOFTEAM Formation, forte de son expérience en Méthodologie, Architecture et Développement, a construit un cursus complet de formation SOA qui vous permet de débiter dès les phases amont, de poursuivre en architecture, et d'aller jusqu'à la réalisation dans le langage de votre choix.

Nouveau catalogue Formation 2009 :

UML pour la maîtrise d'ouvrage	2 j
Analyse et conception avec UML	4 j
SOA Alignement Métier du Système d'Information	2 j
SOA Architecture d'Entreprise (EA)	2 j
SOA Méthodologie pour SOA	2 j
SOA Architecture technique SOA	2 j
SOA Développement de Web Services en Java	3 j
SOA Développement de Web Services en C#	3 j
Architecture distribuée: la synthèse	2 j
Programmation orientée objet avec Java	4 j
Développement d'applications JEE 5	5 j
Développement d'applications JEE 5 Front End	4 j
Développement d'applications JEE 5 Back End avec EJB 3	3 j
Maîtrise du framework (Struts / JSF / Spring / Hibernate)	3 j
Développement d'applications .NET / C#	4 j
Développement d'applications RIA avec	2 j
Web 2.0 (Ajax / Dojo / GWT / FLEX3)	4 j

Convergence SOA, UML2, BPMN, EA

Modélisation EA, BPMN, SOA avec Objectteering SOA Solution	2 j
Analyse et Conception UML2 avec Objectteering Modeler	5 j
Expression de besoins en UML2 avec Objectteering Scope Manager	1 j
Architecture MDA avec Objectteering MDA Modeler	2 j
Génération de code Java, .NET C#, C++ avec Objectteering Developer	1 j



Hibernate : Guide de survie !

Hibernate est un framework de mapping objet-relationnel très populaire mais en fait, mal connu. Pour beaucoup de développeurs, son fonctionnement reste un mystère problématique, source d'exceptions difficilement maîtrisables, générateur d'une activité SQL débordante. Il semble erratique et peu performant. Nous entamons une série d'articles pour le démystifier.

C'est en fait un outil puissant et sûr. Mais comme toutes les belles mécaniques, il faut en connaître les principes fondamentaux pour l'utiliser correctement. Derrière une API simple et une configuration aisément accessible, se cache un moteur complexe que nous allons explorer. Commençons par l'aspect le plus fondamental : la gestion interne des entités.

ENTITÉS ATTACHÉES ET DÉTACHÉES

Toute session Hibernate dispose d'un cache d'entités, appelé cache de premier niveau. Celui-ci est peu connu des développeurs qui utilisent Hibernate et pourtant son rôle est fondamental car toute action Hibernate passe par ce cache. C'est le monde « connu » pour la session :

- Toute entité chargée par la session est conservée dans le cache
- Toute modification apportée à une entité contenue dans le cache sera reportée dans la base de données chaque fois que la session tentera de se « synchroniser » avec elle.

La session Hibernate propose des API comme « save », « saveOrUpdate », « update » ou « delete » et l'on pourrait penser - bien naïvement - que ces API invoquent le moteur de la base de données pour lui transmettre des requêtes de type « Insert », « Update » ou « Delete ». Il n'en est rien. Ces méthodes se contentent (à l'exception de « save » et « saveOrUpdate » qui ont un comportement un peu particulier), d'indiquer au cache que l'entité ciblée doit être créée, mise à jour ou supprimée lors de la prochaine synchronisation. Mais pas avant.

Hibernate, en effet, n'applique de mises à jour sur la base de données que lors des synchronisations (exception faite de « save » et parfois « saveOrUpdate », cf. ci dessus). Avant la synchronisation, Hibernate se contente de noter les modifications à apporter.

Les synchronisations sont déclenchées par un appel à la méthode « flush() » ou lorsqu'une transaction est validée. C'est seulement à ce moment là que la session Hibernate consulte les entités de son cache et applique en base, les mises à jour demandées, dans l'ordre qui lui convient. Cet ordre ne repose évidemment pas sur l'ordre des appels qu'elle a reçus. Notons que seules les entités qui font partie du cache de premier niveau sont concernées par les synchronisations. Les entités qui existent en dehors de ce cache sont ignorées.

Les entités du cache sont appelées entités attachées. Les autres sont dites détachées. Les entités attachées disposent de nombreuses facultés ignorées par les entités détachées. Par exemple, les liens non-initialisés (lazy loading) que l'on tente de traverser seront automatiquement résolus si l'entité est attachée. Si elle ne l'est pas, une exception sera levée. Autre avantage : toute mise à jour sera synchronisée. Inutile donc, d'appeler la méthode « Ses-

sion.update(E) » après modification d'un attribut de « E » si « E » est une entité attachée.

Mais alors, à quoi sert « Session.update() » ? Son rôle est d'attacher une entité qui ne l'était pas en indiquant à la session Hibernate qu'elle contient des modifications qui devront être reportées en base de données lors de la prochaine synchronisation. Son rôle premier est donc d'attacher l'entité, c'est-à-dire, de l'introduire dans le cache. Il n'est pas encore question, à ce moment là, d'invoquer la base de données. C'est la raison pour laquelle « Session.update() » n'invoque pas SQL update.

CONTRAINTES ET VICISSITUDES

Ceci étant dit, est-ce si important, pour développer une application robuste, de comprendre comment Hibernate utilise ce cache ? La réponse est oui. Car ce cache impose deux principes très forts :

- Une entité fonctionnelle (le client « Dupont » de clé primaire « 1 ») ne peut être présent qu'en un seul exemplaire dans le cache. Cela signifie que pour une classe donnée et une valeur de clé primaire donnée, au plus un seul objet Java est référencé par le cache.
- Une entité attachée ne peut être reliée qu'à d'autres entités attachées. On ne peut mélanger entités attachées et détachées. Sinon, le comportement d'Hibernate devient erratique : parfois ce sont des exceptions qui sont émises, d'autres fois, certaines mises à jour ne sont pas effectuées, il est même possible que tout se passe bien ...

Le cache garantit qu'une entité fonctionnelle n'existe qu'en un seul exemplaire en son sein. Ce qui est en soi très intéressant. Si vous tentez d'attacher une entité équivalente à une autre entité déjà présente dans le cache, cela se terminera mal : Hibernate lève une exception. Conjuguée au fait qu'une entité attachée ne peut référencer que d'autres entités attachées, cette faculté peut poser de sérieux problèmes de robustesse ...

Prenons un exemple qui illustre le danger :

Soit une session Hibernate « S » dont le cache est vide. Et soit une grappe d'entités constituée d'un client « C » qui référence à partir de sa collection « C.produits », deux produits « P1 » et « P2 ». « P1 » et « P2 » désignent leur type « TP1 » (« P1 » et « P2 » sont de même type). Seul le contenu de « C » a été modifié.

Cette grappe est détachée. Nous allons l'attacher à notre session. Pour cela nous utilisons la méthode « S.update(C) ; ». Notre objet « C » devient attaché (en précisant au passage qu'une requête SQL Insert doit être invoquée pour C). Mais « P1 », « P2 » ainsi que « TP1 » restent détachés. La seconde règle est donc violée : le comportement d'Hibernate devient aléatoire (fig1) ...

O.K., nous pouvons faire aussi « S.update() » pour les autres



objets de la grappe et résoudre ainsi le problème. Mais cette solution n'est pas vraiment satisfaisante pour deux raisons :

- La première est que cela induit l'émission de requêtes SQL Insert pour « P1 », « P2 » et « TP1 », qui, nous le savons, n'ont pas été modifiées. Exécuter un peu trop de requêtes SQL Select n'est pas forcément gênant. Pour une requête SQL Update c'est probablement plus ennuyeux ...
- Il est souvent très difficile de savoir quels sont les objets présents dans la grappe. Si nous ne connaissons explicitement que « C », comment faire pour savoir si « P1 », « P2 » et « TP1 » sont effectivement là ? Les liens qui unissent ces objets à « C » peuvent être marqués comme « lazy » et ne pas être chargés. Or, le seul fait de les tester provoque une exception en mode détaché (« LazyLoadingException »).

Heureusement, nous avons le « cascading ». Nous pouvons indiquer pour tous nos liens que l'opération SAVE_OR_UPDATE doit être cascadée. Notons que pour que notre application soit robuste, il est nécessaire que TOUTES les relations présentes dans nos entités soient cascadées. A défaut de quoi, il est possible que dans le fin fond d'un graphe d'entités quelques unes d'entre-elles nous échappent ... On comprend qu'un tel mode de fonctionnement risque d'entraîner par effet de bord un flot de requêtes SQL Update absolument incontrôlé !

Mais le pire est à venir ...

Notre grappe d'objets a été récupérée de sessions Hibernate précédentes, maintenant défunes (les entités sont donc détachées). En fait, d'une première session nous avons récupéré « C », « P1 » et « TP1 ». De la seconde session nous avons récupéré « P2 » et à nouveau « TP1 ». Puis nous avons ajouté « P2 » à « C.produits ». Notre graphe d'entités est très similaire à celui présenté plus haut, une seule différence notable : le type de produit est représenté par

deux objets Java distincts (un désigné par « P1 », l'autre par « P2 ») (fig 2).

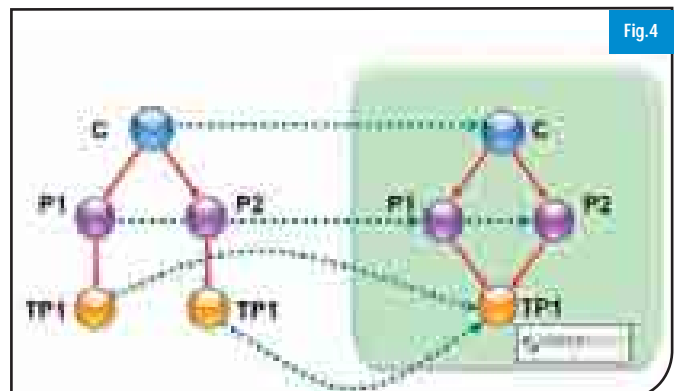
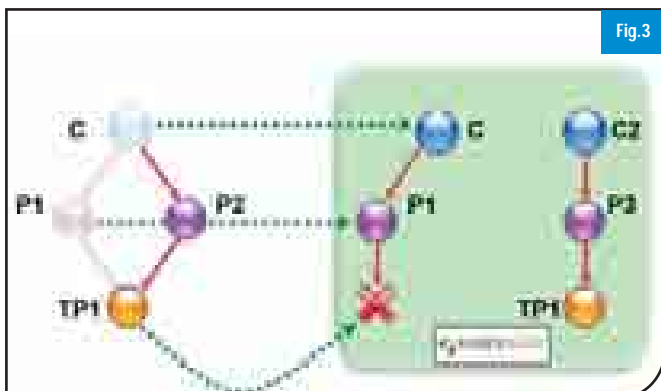
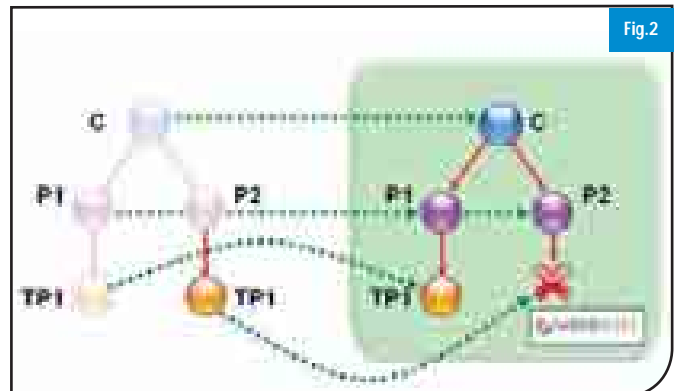
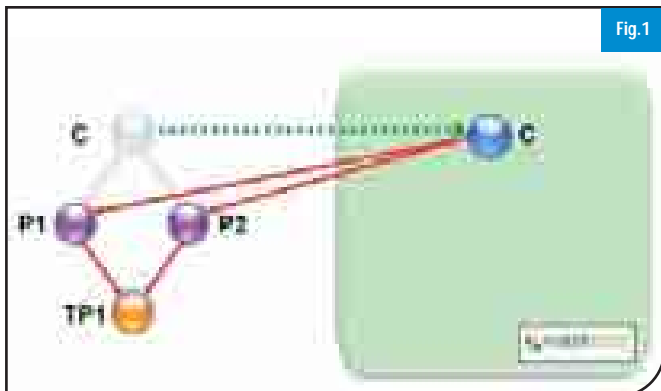
Maintenant, nous faisons « S.saveOrUpdate(C) » ; « C » est d'abord attaché. En cascade, « P1 » est attaché. Ce qui entraîne l'attachement du premier « TP1 ». Puis « P2 » est attaché, et enfin, la session tente d'attacher le second « TP1 ». Et là, c'est le drame car une session Hibernate ne peut contenir qu'un seul exemplaire pour un objet fonctionnel donné !

« Bon, O.K, O.K, » allez vous me dire, « le cascading n'est pas suffisant ». Il faut aussi que le graphe d'objets soit correct. Ce qui est faisable à l'aide d'une programmation rigoureuse. On n'est pas pourtant à l'abri des problèmes. Imaginons que notre session Hibernate ne soit pas vide et qu'elle contienne déjà un autre client « C2 » qui dispose du produit « P3 » qui est aussi de type « TP1 » (fig 4) : il sera impossible d'y insérer notre graphe détaché. Notons que si « P3 » était d'un autre type, rien de fâcheux ne se serait passé. D'où l'impression négative sur la robustesse d'Hibernate que le développeur non averti peut avoir : des fois, ça marche, d'autres fois, non ...

SESSION.MERGE() POUR S'EN SORTIR

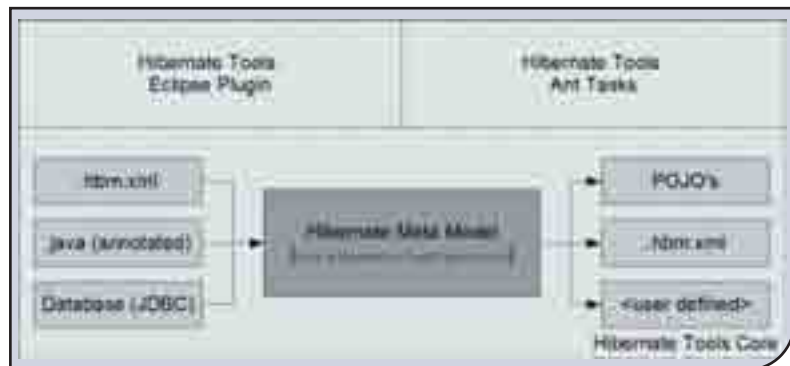
Sommes-nous dans l'impasse ? Heureusement, non : il y a « merge ». « S.merge(E) » est une méthode a priori compliquée. Son utilité ne paraît pas évidente, car son comportement est le suivant :

- Si « E » est déjà attachée, « merge » se contente de retourner immédiatement « E ». Elle ne fait donc rien.
- Si « E » n'est pas attachée, « merge » recherche dans le cache, l'entité fonctionnelle équivalente à « E ». Appelons « E' » cette version attachée de « E ».
- Si « E' » n'est pas trouvée, elle est créée dans le cache (pour une nouvelle entité) ou chargée dans le cache (pour une entité déjà présente en base de données).

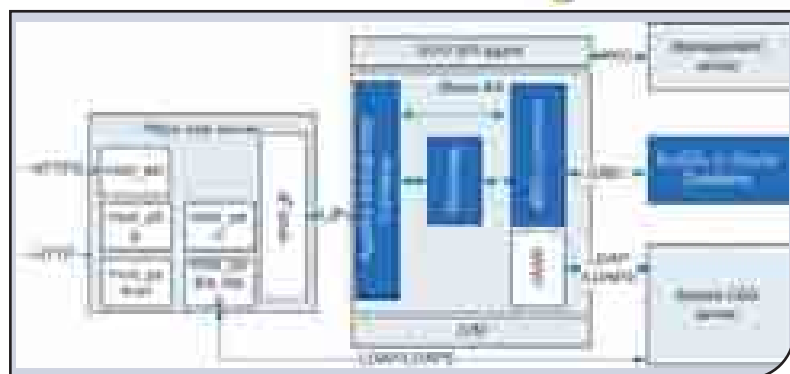




Hibernate est intégré à la plupart des IDE comme dans MyEclipse.



architecture



Hibernate dans l'environnement JBoss

- Le contenu de « E » est copié dans « E' ».
- « merge » retourne « E' ».

Juste une petite précision. Si « E » pointe vers « F », « merge » fera pointer « E' » vers « F' », « F' » étant la version attachée de « F ».

Ce fonctionnement à l'air très complexe, et l'on peut être tenté de se dire : « mon besoin est simple, je n'ai probablement pas besoin de tout cela. Il y a « Save » et « update », faciles à comprendre ». J'espère vous avoir convaincu que les effets de bord de « save », « update » et « saveOrUpdate » sont tels que leur utilisation est vraiment problématique. C'est bien le comportement de « merge » qui est le plus adapté, car plus sûr. Et à l'usage, il se révèle très simple. Explications.

« merge » permet de faire migrer de l'information d'une grappe d'objets détachés vers une grappe d'objets attachés sans faire migrer les objets eux-mêmes d'un environnement à l'autre. Ce qui protège les développements contre tout mélange entre entités attachées et détachées. Comme pour « update », un « merge » peut être cascadié. On peut donc facilement mettre à jour une grappe entière d'objets, mais sans risque cette fois. Pourquoi ? Parce que la condition d'unicité de l'entité dans le cache ne risque pas d'être violée : seul le contenu des entités est transféré.

Notons que « merge » sait aussi bien faire des créations en base que de simples mises à jour. Comme pour « saveOrUpdate », les créations sont immédiates (une requête SQL insert est émise) et les mises à jours simples attendent la prochaine synchronisation. Contrairement à « update », « merge » ne provoque pas de requêtes « SQL update » intempestives. En effet, les entités attachées impliquées par « merge » sont créées de toute pièce (pour matérialiser des créations) ou chargées depuis la base de données. « merge » peut donc savoir immédiatement si les mises à

jour provenant de la grappe des objets détachés modifient réellement les entités. Ce système en revanche génère davantage de requêtes de type « SQL select ». En fait le nombre de requêtes émises par « merge » reste à peu près le même que le nombre de requêtes émises par « saveOrUpdate ». Mais il s'agit de requêtes moins dangereuses et plus efficaces. Afin d'illustrer l'adaptabilité et la robustesse de « merge », reprenons l'exemple précédent en nous mettant dans le cas le moins favorable :

Hors du cache, nous disposons d'une grappe d'objets regroupant le client « C1 », les produits « P1 » et « P2 » et le type « TP1 » matérialisé par deux objets Java, un lié à « P1 », l'autre à « P2 ».

Nous invoquons « merge » sur « C1 ». Le contenu de « C1 » est copié vers le cache. Si la version attachée de « C1 » (déjà présente ou chargée pour l'occasion) est identique à la version détachée, aucune mise à jour ne sera ensuite demandée. Il en va de même pour « P1 » et « P2 ». « TP1 » sera copié deux fois. Ce n'est pas idéal, mais c'est déjà mieux qu'une exception systématique.

Mais il y a mieux : nous ne sommes plus obligés de généraliser l'opération « merge » sur l'ensemble de la grappe. Nous pouvons par exemple ne pas procéder au « cascading » entre produits « P1 » et « P2 » d'une part et « TP1 » d'autre part. Ce qui est particulièrement indiqué si « TP1 » est un objet de référence qui n'évolue presque jamais.

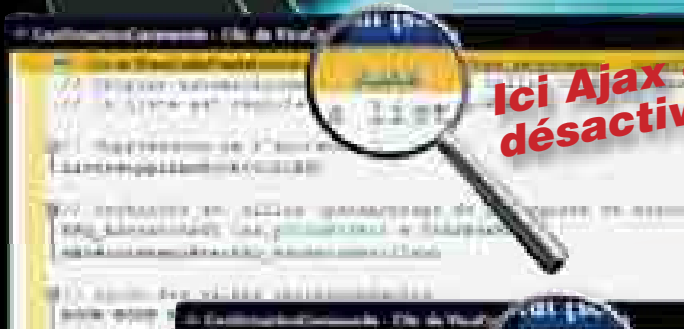


■ Henri Darmet

Directeur Technique - Objet Direct / Homsys Group
Objet Direct, filiale à 100% de Homsys Group est une société de conseil, de services et de formation, spécialisée sur les technologies objet et Web. Conseil en méthodologie, en architecture et en urbanisation du SI, développement applicatif, édition et distribution de logiciels. www.objetdirect.com

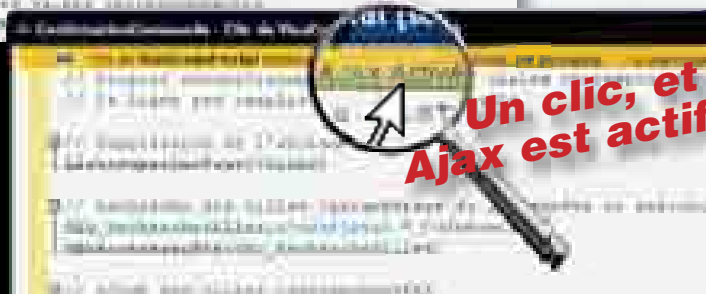
WEB 2.0

Ajax en 1 clic avec WEBDEV 14



Ici Ajax est désactivé

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



**Un clic, et
Ajax est actif**

WEBDEV 14 permet de développer jusqu'à 10 fois plus vite tous les types de sites et d'applications reliés aux données de votre entreprise. L'activation d'**AJAX** dans vos sites s'effectue naturellement: un simple clic

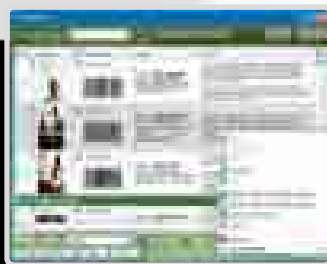
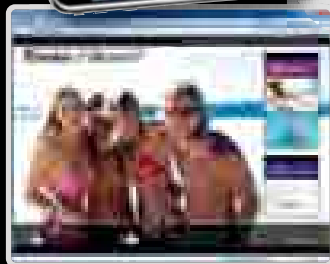
dans l'éditeur de code indique que le code à exécuter est de type «**Ajax**».

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

AGL complet: langage L5G, RAD PHP, débogueur, Web Services, 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 14
(logiciel **professionnel**, dossier gratuit et témoignages -200 pages- sur simple demande)

WEBDEV 14
est compatible
avec
WINDEV 14



Un des nombreux
exemples livrés avec
WEBDEV : portail
Intranet **Honolulu**,
à télécharger gratui-
tement (sur pcsoft.fr).

www.pcsoft.fr

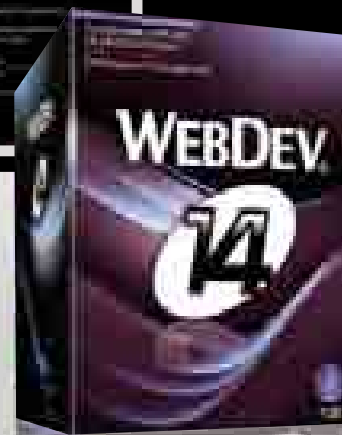
info@pcsoft.fr

Dossier technique et témoignages sur simple demande

Tél Province **04.67.032.032** Tél Paris **01.48.01.48.88**

UN CODE MULTI-PLATEFORMES : WINDOWS, WEB, MOBILE

WEBDEV®



Fournisseur Officiel de la
Préparation Olympique



Partitionnement avec MySQL 5.1

L'une des fonctionnalités phare de MySQL 5.1, est la possibilité de partitionner ses tables. Certes il était déjà possible, depuis la v3.23, de diviser ses tables "manuellement" avec le moteur de stockage MRG_MYISAM, certes la technologie partitionnement (algorithme KEY) existe depuis la 4.1 avec le moteur NDB_Cluster, mais découper ses tables simplement, par "range", "list", "hash" ou "key", fusionner les partitions, les séparer, les sous-partitionner... tout ceci n'est vraiment possible que depuis la 5.1.

La version de MySQL utilisée pour cet article est la 5.1.30 Community Server, version Windows. Le partitionnement est un type d'architecture qui consiste à diviser une table (ou une base, ndlr) en plusieurs parties. Cette technique de conception a deux principaux objectifs :

- **Augmenter les performances** : partitionner est d'autant plus utile que le volume de données est important. On peut voir les partitions comme des sous-ensembles d'une même table, l'optimiseur de MySQL sait sur quelles partitions chaque enregistrement se trouve. Il se peut alors, si les données recherchées ne concernent pas toutes les partitions, que le temps de réponse d'une recherche s'en trouve amélioré. Ce mécanisme s'appelle le PRUNING, en d'autres termes, c'est la faculté d'analyser seulement les partitions qui contiennent les enregistrements recherchés.
- **Simplifier certaines tâches de maintenance des données** : si des données obsolètes se trouvent dans une même partition, effacer entièrement cette partition (DROP PARTITION) est plus rapide que d'effacer tous les enregistrements les uns après les autres (DELETE FROM). De plus, contrairement au moteur de stockage MERGE qui permet de faire du partitionnement "manuel" depuis la version 3.23.25, MySQL 5.1 permet de partitionner et de maintenir beaucoup plus simplement ses partitions.

Partitionner une table

Le partitionnement se définit au niveau de la structure de la table, lors de la création de cette dernière (CREATE TABLE) ou alors après, en modifiant la structure de la table (ALTER TABLE). MySQL propose quatre types de partitionnement :

- **RANGE** : Permet de spécifier des intervalles de valeurs
- **LIST** : division des données sous forme de listes de valeurs
- **HASH** : utilisation d'une clé de hashage pour répartir les données de façon homogène
- **KEY** : utilisation également d'une clé de hashage pour répartir les données de façon homogène.

La syntaxe SQL de MySQL 5.1 a été enrichie pour l'implémentation du partitionnement avec l'apparition du mot clé (partition). Les clauses permettant de partitionner une table doivent être placées à la fin du code SQL de création de la table, de plus chaque type de partition à sa syntaxe propre (Listing 1)

Listing 1. Syntaxe de création d'une table partitionnée

```
CREATE TABLE nom_de_la_table ( <definition_de_la_table> )
```

```
ENGINE = <moteur_de_la_table>
[definition_de_la_partition]
```

```
[definition_de_la_partition]
PARTITION BY
    { [LINEAR] HASH(expr)
    | [LINEAR] KEY(column_list)
    | RANGE(expr)
    | LIST(expr) }
[PARTITIONS nombre]
```

Les contraintes

La clé de partition doit être de type entier ou composée d'une fonction qui retourne un entier ou une valeur NULL, cette limitation ne s'applique pas pour le partitionnement par KEY.

Si la table à un index unique ou une clé primaire, la clé de partition doit être au moins une partie de cet index unique (ou clé primaire) (Listing 2). Les tables de type **merge**, **federated**, **cvs** ne peuvent être partitionnées.

Listing 2. la clé de partition doit être au moins une partie de la clé primaire

```
CREATE TABLE commande (
    ID_commande int NOT NULL AUTO_INCREMENT,
    ID_client smallint NOT NULL,
    PRIMARY KEY (ID_commande)
) ENGINE=MyISAM
PARTITION BY RANGE (ID_client) (
    PARTITION p0 VALUES LESS THAN (16384),
    PARTITION p1 VALUES LESS THAN (32768)
);
ERROR 1503 (HY000): A PRIMARY KEY must include all columns in
the table's partitioning function
```

Le partitionnement par RANGE

La clause de partitionnement est PARTITION BY RANGE (colonne). Il faut ensuite donner un nom à la partition : PARTITION nom_partition et définir la borne maximale (non incluse) de l'intervalle : VALUES

LESS THAN (borne_max_partition). La borne maximale de toutes les partitions peut être définie par le mot clé: MAXVALUE. (Listing 3).

Listing 3. Table partitionnée par range

```
CREATE TABLE employe_range (
  ID tinyint unsigned NOT NULL,
  Name char(35) NOT NULL DEFAULT '',
  PRIMARY KEY (ID)
) ENGINE=MyISAM
PARTITION BY RANGE (ID) (
  PARTITION p_id_moins_de_64 VALUES LESS THAN (64),
  PARTITION p_id_moins_de_128 VALUES LESS THAN (128),
  PARTITION p_id_moins_de_192 VALUES LESS THAN (192),
  PARTITION p_id_moins_de_256 VALUES LESS THAN (MAXVALUE)
);
```

Tout ID strictement inférieur à 64 est stocké dans la partition *p_id_moins_de_64*, Tout ID strictement inférieur à 128 est stocké dans la partition *p_id_moins_de_128*, etc.

L'ordre des partitions est important, la clause MAXVALUE ne peut être que la dernière de la liste et les bornes de partitions ne peuvent être définies que de la plus petite à la plus grande.

Le plus important dans le partitionnement est le choix du critère de partitionnement, c'est-à-dire la colonne choisie pour partitionner la table. Ce choix se fait principalement en fonction du type de requêtes exécutées sur la table. L'algorithme *range* est très utile pour partitionner des dates. Mais les types *datetime*, *timestamp* de MySQL ne sont pas des entiers. Il faut donc utiliser une fonction sur cette colonne qui renvoie un entier ou la valeur NULL. Le tableau 1, liste quelques unes des fonctions *date* qui retournent un entier.

La fonction **YEAR()** est toute indiquée, elle extrait l'année d'une date et la retourne sous la forme un entier (Listing 4). Une autre fonction recommandée est **TO_DAYS()**. Elle prend une date en entrée et retourne un entier qui représente le nombre de jours depuis l'année 0. La fonction peut également être mise lors de la définition de chaque partition. A noter que la valeur NULL est considérée comme étant inférieure à tous les nombres entiers, en d'autres termes si le champ a des valeurs NULL, ces dernières iront dans la première partition.

Listing 4. Partition par année avec la fonction year

```
CREATE TABLE inscription (
  ID mediumint unsigned NOT NULL AUTO_INCREMENT,
  date_inscription datetime,
  KEY (ID)
) ENGINE=MyISAM
PARTITION BY RANGE (YEAR(date_inscription)) (
  PARTITION p1 VALUES LESS THAN (1970),
  PARTITION p2 VALUES LESS THAN (1980),
  PARTITION p3 VALUES LESS THAN (1990),
  PARTITION p4 VALUES LESS THAN (2000),
  PARTITION p5 VALUES LESS THAN (2010)
);
```

Le partitionnement par LIST

La clause de partitionnement est **PARTITION BY LIST** (colonne). Il faut ensuite donner un nom à la partition: **PARTITION nom_partition**

SGBD> MySQL

Formation MySQL DBA



"Montez en puissance sur l'administration MySQL et préparez vous à passer la certification"

Au Programme des 5 jours :

- * Installer et configurer MySQL
- * Créer une base de données optimale selon le contexte
- * Superviser / monitorer un serveur MySQL
- * Gérer l'intégrité des données
- * Gérer la sécurité des serveurs MySQL
- * Utiliser toutes les fonctionnalités de MySQL 5
- * Scalabilité de MySQL

Prochaines sessions

Paris 2009		Lyon 2009		Tarif
09/03/2009	30/03/2009	09/03/2009	20/04/2009	1999 € HT
20/04/2009	11/05/2009	08/06/2009	06/07/2009	
08/06/2009	27/07/2009			

LE SPECIALISTE DE LA FORMATION POUR L'OPEN SOURCE

Informations

01 45 28 09 82

www.anaska.com





et définir la liste des valeurs pour chaque partition: **VALUES IN** (valeur1, valeur2,...,valeurN).

L'algorithme LIST permet de spécifier la valeur NULL comme valeur de partitionnement. Par contre, si vous essayez d'insérer un nombre qui n'est pas dans la liste des valeurs permises d'une des partitions, une erreur sera lancée par MySQL. Ce mécanisme renforce la cohérence des données dans le SGBDR (Listing 5).

Listing 5. Table partitionnée par list avec valeur NULL

```
CREATE TABLE short_message_list (
  id_message int unsigned NOT NULL AUTO_INCREMENT,
  nature_message tinyint unsigned,
  KEY (id_message)
) ENGINE=Memory
PARTITION BY LIST (nature_message) (
  PARTITION p_undef VALUES IN (NULL),
  PARTITION p_faux VALUES IN (0),
  PARTITION p_vrai VALUES IN (1)
);

INSERT INTO short_message_list (nature_message) VALUES (0);
Query OK, 1 row affected (0.00 sec)

INSERT INTO short_message_list (nature_message) VALUES (NULL);
Query OK, 1 row affected (0.00 sec)

INSERT INTO short_message_list (nature_message) VALUES (42);
ERROR 1526 (HY000): Table has no partition for value 42
```

Le partitionnement par HASH

La clause de partitionnement est **PARTITION BY HASH** (colonne). Il faut ensuite donner le nombre de partitions souhaitées: **PARTITION nombre_partition**. (Listing 6).

L'algorithme de partitionnement hash, idéal pour les valeurs séquentielles, comme les **AUTO_INCREMENT**, permet de distribuer de façon homogène les données dans les différentes partitions créées. La distribution des données sur les partitions est assez simple, c'est du modulo (le reste de la division). Si vous avez 3 partitions, p0, p1, p2, un enregistrement ira dans p0, le suivant dans p1, le suivant dans p2, le suivant dans p0, etc.

Pour être plus précis, le calcul fait par MySQL est le suivant :

```
IF(ISNULL(valeur_partition), 0, ABS(valeur_partition)) MOD nbr_de_partitions
```

L'algorithme hash, a une variante, **LINEAR HASH**, qui a pour effet de rendre plus rapides les tâches de maintenance sur les table partitionnées (suppression, fusion, ajout...), mais en contrepartie, la distribution des données sur les différentes partitions est moins homogène.

Listing 6. Table partitionnée par hash

```
CREATE TABLE ville_fra (
  id_ville mediumint unsigned NOT NULL AUTO_INCREMENT,
  nom char(35) NOT NULL DEFAULT '',
```

```
Population int unsigned NOT NULL DEFAULT '0',
PRIMARY KEY (id_ville)
) ENGINE=Archive
PARTITION BY HASH (id_ville) PARTITIONS 4 ;
```

Le partitionnement par KEY

La clause de partitionnement est **PARTITION BY KEY** (colonne1, colonne2,...), Avec zéro ou plusieurs colonnes. Il faut ensuite donner le nombre de partitions souhaitées: **PARTITION nombre_partition**. L'algorithme de partitionnement key est proche de celui du hash, il permet lui aussi de distribuer de façon homogène les données dans les différentes partitions créées. Il se différencie néanmoins par la possibilité donnée à l'utilisateur de ne pas choisir de colonne (Listing 7) ou alors d'en spécifier plusieurs, à condition qu'elles appartiennent toutes à la clé primaire (ou un index unique), ou alors que la table ne contienne pas de clé primaire ou d'index unique (Listing 8).

Listing 7. Partitionnement par key sans spécifier de colonnes

```
CREATE TABLE livre3 (
  isbn char(13) NOT NULL,
  titre char(255) NOT NULL DEFAULT '',
  auteurs tinyint unsigned NOT NULL,
  PRIMARY KEY (isbn, titre)
) ENGINE=MyISAM
PARTITION BY KEY () PARTITIONS 7 ;
Query OK, 0 rows affected (0.09 sec)
```

Listing 8. Partitionnement par key multi-colonne

```
CREATE TABLE livre2 (
  isbn char(13) NOT NULL,
  titre char(255) NOT NULL DEFAULT '',
  auteurs tinyint unsigned NOT NULL,
  PRIMARY KEY (isbn)
) ENGINE=MyISAM
PARTITION BY KEY (isbn, titre) PARTITIONS 5 ;
ERROR 1503 (HY000): A PRIMARY KEY must include all columns in the table's partitioning function

CREATE TABLE livre4 (
  isbn char(13) NOT NULL,
  titre char(255) NOT NULL DEFAULT '',
  auteurs tinyint unsigned NOT NULL,
  KEY (isbn)
) ENGINE=MyISAM
PARTITION BY KEY (isbn, titre) PARTITIONS 5 ;
Query OK, 0 rows affected (0.09 sec)
```

Il y a encore beaucoup de choses à dire sur les partitions. Mais tout ceci sera pour un prochain numéro.

■ Olivier DASINI

Consultant et formateur officiel certifié MySQL. Responsable du pôle SGBDR chez Anaska Alter Way Group. Auteur de nombreux articles sur MySQL, il est également co-fondateur du MySQL User Group Francophone: LeMUG.fr

Mobiles : développez 10 fois plus vite avec WINDEV Mobile 14

VERSION
EXPRESS
GRATUITE
Téléchargez-la !



Les **codes-barres** sont
gérés en
standard

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

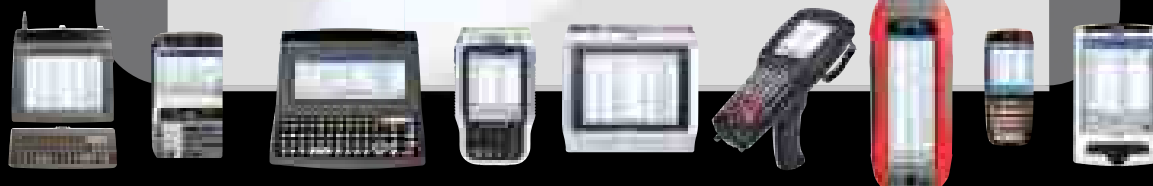
La **puissance** et la **facilité** de développement de WINDEV Mobile 14 permettent un développement complet en quelques journées.

Déploiement **gratuit** sans redevances (base de données incluse), réplication, WiFi, 3G, Internet, lien avec votre S.I., ... tout est inclus.

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

(Logiciel **professionnel**, dossier gratuit +100 témoignages sur simple demande. Version Express gratuite en téléchargement libre sur www.pcsoft.fr)

WINDEV
Mobile 14
est compatible
avec
WINDEV 14 et
WEBDEV 14



Dossier technique et témoignages (200 pages +DVD) sur simple demande
Tél Province **04.67.032.032** Tél Paris **01.48.01.48.88** info@pcsoft.fr

UN CODE MULTI-PLATEFORMES : WINDOWS, WEB, MOBILE

WINDEV® Mobile



www.pcsoft.fr



Intégration de données avec SQL Server Integration Services

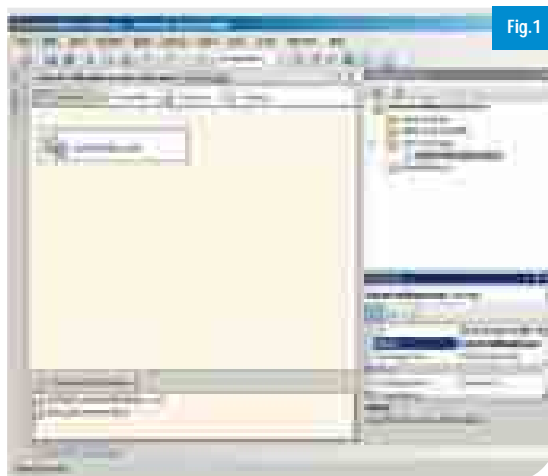
Dans le monde de l'ETL, Microsoft nous avait proposé une réelle avancée avec la version 2005 de son produit maison SSIS (SQL Server Integration Services). Né des cendres de DTS (Data Transformation Services), version précédente de l'outil d'intégration de Microsoft présent dans SQL Server 2000, SSIS offrait une interface de développement riche intégrée à Visual Studio. Outre un environnement de développement unifié et une facilité de prise en main, SSIS élargissait la liste des composants disponibles pour les traitements des flux tout en se montrant un outil robuste, fiable et performant.

Avec SQL Server 2008 et la nouvelle mouture de SSIS, le bond en avant est significatif même s'il sera bien moins grand qu'à la sortie de SSIS 2005. Parmi les nouvelles fonctionnalités présentes, il en est une qui retiendra particulièrement l'attention des acteurs et spécialistes de l'ETL : le DataProfiling Task. Avant d'entrer en détail sur le composant, rappelons tout d'abord ce qu'est le DataProfiling.

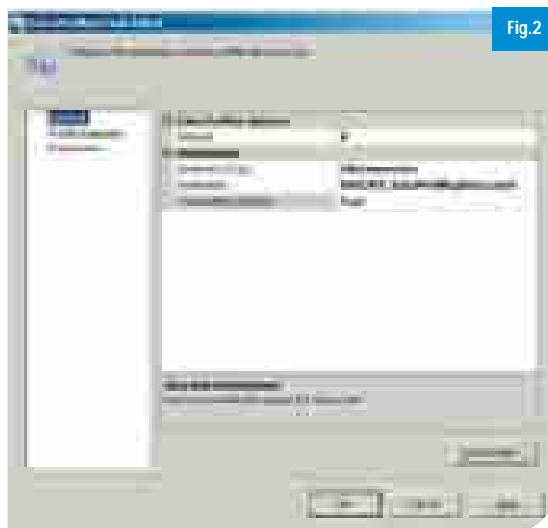
Vous avez dit DataProfiling ?

Pour résumer, le DataProfiling est la méthode permettant d'accéder aux données sources d'un point de vue analytique afin de valider leurs qualités mais également d'en extraire des éléments d'analyse.

Le DataProfiling construit un inventaire exhaustif des données sources dans le but de fournir et d'intégrer des données de meilleure qualité dans le datawarehouse. Il est un des points de focus les plus importants de tout projet ETL/décisionnel. C'est à juste titre que le Kimball Group, acteur majeur de la méthodologie inhérente au projet de datawarehouse, l'inclut dès les premières phases d'un projet en le considérant comme le numéro un des 34 sous-systèmes, best-practices, qu'il a défini. En effet, l'analyse des données sources dès les premières heures de vie d'un projet met en lumière de potentiels problèmes. Le DataProfiling délivre éga-



Composant DataProfiling Task (SSIS 2008)



lement des statistiques et déduit des hypothèses sur les données à intégrer. Ainsi, une connaissance et une maîtrise parfaite des données sources autorisent le développement de solutions ETL plus fiables, plus

DataProfiling Task Editor : Choix de la destination

robustes et évitent que ne surviennent de mauvaises surprises lors de la phase de conception.

Enfin, le DataProfiling relève d'éventuels changements dans les données sources. En effet, un ETL peut avoir été conçu sur la base d'une documentation, d'une étude des données à un instant T. Or, imaginons que les données sources changent pour des raisons fonctionnelles et/ou techniques. La documentation étant souvent le parent pauvre d'un projet, il se peut que la documentation d'origine ne reflète pas les évolutions intervenues ou encore que l'équipe en charge du développement de l'ETL n'ait pas été informée des changements. Le DataProfiling en temps réel des données sources permettra alors de mettre en évidence les mises à jour du système source et l'adéquation de l'ETL. Laissons désormais de côté le caractère théorique du DataProfiling pour la mise en pratique via SSIS 2008.

Utiliser le DataProfiling

Le DataProfiling de SSIS 2008 se compose de 2 éléments :

- la tâche DataProfiling Task disponible au niveau du control flow,
- le DataProfileViewer pour visualiser les fichiers XML générés en sortie par la tâche DataProfiling Task. Il se situe, en dehors du flux, dans le répertoire d'installation de SQL Server.

Abordons tout d'abord le composant.

[Fig.1]



La première des configurations à effectuer est de choisir le type de sortie désiré : à savoir, charger les données résultant du profiling dans un fichier XML, ou stocker ces informations au sein d'une variable préalablement définie au niveau du package SSIS. Pour cette démonstration, et afin de pouvoir illustrer DataProfileViewer nous choisirons une sortie de type fichier XML.

Nous pouvons à ce stade utiliser une connexion déjà existante à un fichier XML au niveau des connexions managers du package SSIS, ou en spécifier une nouvelle au niveau de la tâche qui viendra ensuite s'ajouter à la liste de celles du package SSIS. [Fig.2]

Deuxième étape, le choix du type de profiling désiré et la source à laquelle l'appliquer.

Concernant les possibilités de profiling disponibles nous avons à notre disposition :

- Candidate Key Profile Request
- Column Null Ratio Profile Request
- Column Statistics Profile Request
- Column Value Distribution Profile Request
- Column Length Distribution Profile Request
- Column Pattern Profile Request
- Functional Dependency Profile Request
- Value Inclusion Profile Request

Pour cette démonstration, nous optons pour un profiling de type " Column Statistics Profile Request " que nous appliquerons aux heures d'absences pour maladie présentes dans la table [HumanResources].[Employee] de la base exemple AdventureWorks.

A ce stade, les seules connexions managers proposées sont celles de type ADO.NET ; Microsoft ayant décidé avec SSIS 2008, de mettre en avant ADO.NET afin d'optimiser l'intégration aux systèmes et le support d'applications tierces.

Il est possible de sélectionner une collection de colonnes destinées au profiling. [Fig.3]

Analysons le fichier XML produit en

sortie par le DataProfiling Task à l'aide du DataProfileViewer présent dans :

```
%programfiles%\Microsoft SQL Server  
\100\DTS\Binn\DataProfileViewer.exe
```

Comme nous pouvons le constater, le profiling donne accès à des données statistiques. Plusieurs scénarii s'offrent à nous pour en démontrer l'utilité. Par exemple :

- imaginer de tracer au fil du temps ces statistiques pour en mesurer l'évolution, décider, au niveau du flux SSIS, des différentes actions en fonction des valeurs qui entrent dans le flux en les comparant aux données statistiques du profiling,
- rediriger telle ou telle ligne si elle se situe au dessus de la moyenne et ainsi de suite. [Fig.4] ...

Lançons de nouveau le package SSIS contenant la tâche de DataProfiling en choisissant, cette fois, un profiling de type " Column Length Distribution Profile Request " sur le type d'offres spéciales présentes dans la table [Sales].[SpecialOffer].

Le résultat obtenu, ouvert avec DataProfileViewer, est le suivant : [Fig.5]

Nous constatons, de nouveau, l'utilité des statistiques renvoyées à la prise de décisions pour, par exemple :

- affiner le datatype de la colonne dans le datawarehouse qui recevra ces données grâce à l'indication sur la longueur maximale des données contenues en source.
- connaître le nombre de valeurs distinctes en source et, éventuellement, déduire une dimension.

...

Ce type de profiling peut également amener à mettre en avant des problèmes de valeurs invalides. Par exemple, si pour une colonne nous ne devrions avoir que des références de produits sur 8 caractères et que le profiling nous renvoie des statistiques sur des valeurs de plus (ou moins) 8 caractères.

A noter qu'en double-cliquant sur l'une des lignes statistiques, un aperçu des données concernées s'ouvre dans la partie inférieure du viewer.



Fig.3
Data Profiling Task Editor : Choix du type de profiling et source d'analyse



Fig.4
Data Profile Viewer : Column Statistics Profile



Fig.5
Data Profile Viewer : Column Length Distribution Profile

Allez-y !

Les cas présents mettent en avant toutes les perspectives offertes par le DataProfiling de SSIS 2008. Nul doute que l'exploitation de ces données permettra au travers de SSIS 2008 de proposer des ETL orientés metadata plus complets. En élargissant ainsi la connaissance des données à intégrer, les développements n'en seront que plus aisés, fiables et robustes.

Une démarche à adopter sans modération !

■ Bertrand Renotte

Ingénieur Systèmes, Neos-SDI



Données géospatiales dans SQL Server 2008

C'est la grande nouveauté de SQL Server 2008 pour les développeurs : la possibilité de stocker dans la base de données des "polygones" apposés sur un plan ou une sphère, et surtout de pouvoir les exploiter en masse ... quelques exemples pratiques :

- La plus petite distance entre un point et un ensemble d'autres points (le restaurant, la canalisation d'eau, le prochain client d'une tournée) les plus proches ? le parcours le moins énergivore pour un avion ?
- L'aire totale d'un polygone ou ensemble de polygones (aire déclarée pour un terrain ? statistique moyenne par m2 par habitant selon quartier ? Rentabilité de ventes selon surface consacrée par typologie de produits ?)
- Les polygones à moins d'une distance définie d'un autre polygone (ensemble de bâtiments impactés par des travaux ? annonces immobilières pouvant également intéresser le client ?)
- Les intersections de polygones (chemin le plus court d'un point à un autre ?)

Pour cela : 2 nouveaux types de données :

Type	Description	Exemple
Geometry	Position sur une surface plane (X,Y) Pas de notion d'altitude (Z)	declare @gm geometry set @gm = geometry::STGeomFromText('POINT(10 10)', 0)
Geography	Position sur une sphère (latitude, longitude) Pas de notion d'altitude non plus (Z)	declare @gg geography set @gg = geography::Parse('POINT(47.6456 -122.12551)')

Et 4 types de polygones :

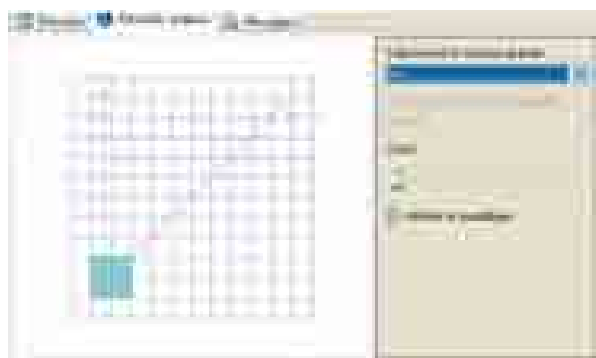
Type	Description	Résultat
Point	POINT(1 1)	
Trait	LINESTRING(4 4, 6 6)	
Polygone simple	POLYGON((0 0, 0 4, 4 4, 4 0, 0 0))	
Polygone à trous	POLYGON (((6 6, 10 6, 10 10, 6 10, 6 6), (7 7, 7 8, 8 7, 7 7), (9 9, 9 8, 8 9, 9 9)))	

Ces nouveaux types de données viennent également avec leur nouveau type d'index qui va permettre d'assurer des temps de réponses optima : les index spatiaux. Ceux-ci définissent un espace limité (la "bounding box") et un ensemble de 4 niveaux de grilles ("grids") successifs qui va permettre de trouver rapidement le polygone cherché qui est lui, référencé comme contenu dans tel et tel carré de chacune des grilles.

<pre>CREATE SPATIAL INDEX ix_lieu ON MaCarte(lieu) WITH (BOUNDING_BOX = (0, 0, 500, 500), GRIDS = (LOW, LOW, MEDIUM, HIGH), CELLS_PER_OBJECT = 20)</pre>	
--	--

Lorsqu'une recherche est faite : SQL Server utilise dans un premier temps cet ensemble de grilles pour déterminer les polygones pouvant potentiellement répondre à la demande, puis dans un second

temps, écarte les polygones ne correspondant finalement pas en appliquant réellement les formules mathématiques sur les polygones restants. Le fait de gérer des données spatiales pose également une question cruciale : comment tester/visualiser ces données sans avoir à faire un développement spécifique ? SQL Server Management Studio inclut désormais un onglet "Résultats spatiaux" qui affiche graphiquement le résultats des requêtes "spatiales". Une autre alternative gratuite existe avec GeoQuery 2008 qui permet notamment d'appliquer votre propre fond de carte et/ou de s'interfacer avec celle de Google Maps ou de Virtual Earth.



Dernier détail important : ces données spatiales ne sont que des données 2D (et non 3D, hormis le point qui exceptionnellement peut avoir 3 dimensions et une mesure associée, mais qui ne sont pas indexés).

■ Patrice BERQUET, Groupe SQLI - Architecte DBA Microsoft

SGBD géospatiale : les critères

Aujourd'hui de nombreuses bases de données supportent les données géospatiales et savent, plus ou moins bien, les traiter, les transformer, les manipuler, les afficher. On peut citer en vrac : SQL Server 2008, MySQL, Oracle, PostgreSQL, IBM DB2 ou Ingres. Ces fonctions sont natives à la base ou par des modules additionnels. Il y a quelques mois Linagora avait établi un rapport complet sur 3 SGBD spatiaux (étude des bases de données spatialisées, pour le CNES) : Oracle Spatial / Locator, MySQL et PostGIS. On parlera souvent pour qualifier ces bases de données de GIS (Geographic information system).

Plusieurs points sont à considérer pour choisir un SGBD pour des traitements géospatiaux :

- support de OGC : spécification de l'Open Geospatial Consortium. Définit les usages et les fonctions pour gérer les données spatiales. Vérifie le niveau de support.
- index spatiaux : point sensible car il facilite la requête de ces données et offre de bonnes performances. A préférer aux index "normaux".
- présence d'outils dédiés : par exemple DB2 inclut des outils de visualisation et d'interrogation.
- gestion des coordonnées et données géocentriques
- Support et gestion de SRS (système de référence spatial) et des prédicats spatiaux. Les prédicats sont des fonctions / opérateurs pour tester les relations spatiales entre les différents objets...

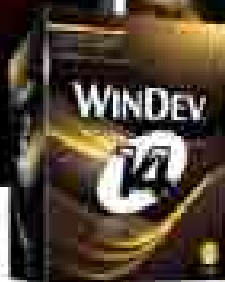
Pour aller plus loin : http://www.bostongis.com/PrinterFriendly.aspx?content_name=sqlserver2008_postgis_mysql_compare

■ F.T.

VOUS AUSSI, DÉVELOPPEZ 10 FOIS PLUS VITE



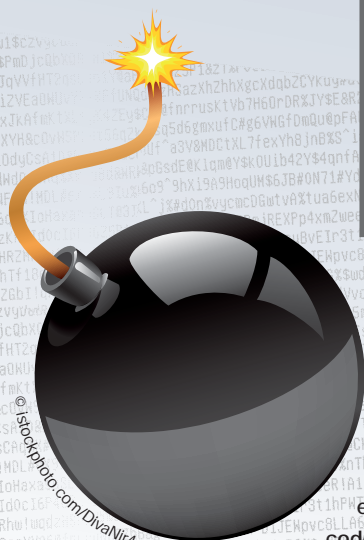
DEMANDEZ LE DOSSIER GRATUIT : 244 pages +
DVD + Version Express gratuite + 112 Témoignages.
Tél: 04.67.032.032 - 01.46.01.48.88 Mail: info@pcsoft.fr



WINDEV : eh! «Langage le plus productif du marché»
www.pcsoft.fr

Hacking

Sécuriser votre code



Programmez ! vous met régulièrement en garde contre les failles de sécurité, le code mal ficelé, les erreurs de programmation. Aujourd'hui, la situation n'est pas pire qu'il y a quelques années mais les effets n'en sont pas moins

catastrophiques. Pourtant, pour sécuriser une application, le développeur n'a, finalement, que peu de choses à faire. Il y a des réflexes à systématiser pour éviter les failles les plus basiques. Combien de sites web, d'applications oublient encore de vérifier le bon format de saisie d'un champ, en entrée et en sortie ? Or, ce petit oubli peut mettre K.O. votre site et si le hacker est bon, c'est le serveur qui tombe... Nous verrons aussi pourquoi il peut être utile de hacker son propre code ! Rien de malicieux là-dedans, juste des techniques peu connues pour comprendre un bug dans son code que l'on n'arrive pas à résoudre...

Il y a quelques semaines, un rapport sur les 25 erreurs de code les plus fréquentes a eu

le mérite de rappeler à tous

les développeurs les risques d'attaques, faute d'un code mal testé, mal sécurisé.

Reste à savoir si ces

préconisations seront suivies par les éditeurs, les entreprises, les développeurs. Mais au final, qui est responsable d'un mauvais code dans un logiciel vendu dans le commerce ou sur un site web commercial ? Cette question légale n'est pas anodine et personne ne souhaite réellement ouvrir un débat explosif pour tout le monde. Faut-il établir une norme ISO officielle ?

UN SURCÔÛT DE 10 À 15%

On estime le surcoût d'un code sécurisé à environ 10-15 %, une des questions étant de savoir qui va payer. Et si tout le monde accueille avec bienveillance ce rapport, ce n'est pas pour autant que les acteurs du monde informatique bougent !

Au-delà du simple aspect programmation, le code sécurisé doit devenir un réflexe et il doit même être intégré dans le cycle de développement de tout projet. C'est ce que l'on appelle en anglais : Security Development Lifecycle (ou SDL). Que vous soyez dans une SSII, indépendant, ou simplement passionné, vous aurez toujours quelque chose à apprendre sur la sécurité. Et n'oubliez jamais : on ne comprend pas les contre-mesures si on ne comprend pas les techniques d'attaques, les failles et leurs mécanismes. Dans ce dossier, nous allons aborder de nombreux aspects du hacking : décompilation, failles, attaques... Et ce, sur différents langages et contextes. Nous avons fait intervenir nos meilleurs experts et notamment un des hackers bien connus de la scène française : ivanlefou.

Dans les prochains numéros, nous aborderons des sujets qui n'auront pas été présents dans ce dossier. 2009, année du hacking pour les développeurs ?

■ François Tonic



“Hack your code”

Il est toujours utile d'avoir plusieurs outils à sa disposition pour résoudre un problème. Lorsqu'on est développeur, on se rend souvent compte que des techniques non-traditionnelles peuvent être efficaces dans des cas particuliers. Cet article a pour but de présenter différentes méthodes qui sont utilisées dans le monde de la sécurité informatique et qui s'appliquent potentiellement au développement traditionnel.

Quand on parlera de "hack" dans cet article, il s'agira de techniques non-standard, c'est-à-dire qu'on ne trouve ni dans les manuels ni dans les bonnes pratiques de code. En aucun cas nous ne parlerons de méthodes malveillantes qui visent à corrompre un système.

Cet article sera surtout orienté développement sous Windows, allant aussi bien de Windows 2000 jusqu'à Windows 7 en englobant bien sûr les architectures x86 et x64. Il faut bien comprendre que tout ce que nous allons évoquer ici peut se porter sur d'autres systèmes, même si les outils et les termes sont différents, au final les systèmes se ressemblent beaucoup. Les Linux-addicted devraient être donc en mesure de suivre cet article :). Celui-ci s'oriente plus vers des développeurs d'applications compilées, celles qui sont interprétées par le CPU (comme des programmes codés en C ou C++). Les personnes développant dans des langages nécessitant une machine virtuelle ou un interpréteur (Java ou .Net ou encore les langages de scripts tel que Python et Ruby) ne trouveront peut être pas forcément leur bonheur.

Pourquoi hacker son code ?

On peut se demander quel est l'intérêt de maîtriser des méthodes obscures provenant du monde du "hack". Il arrive parfois qu'on se retrouve dans une situation bloquante, que la documentation ne prévoit pas. Une recherche sur le net n'aboutit pas toujours : le problème étant trop simple à décrire, nous nous retrouvons noyés dans les résultats, ou alors le problème est trop pointu pour obtenir des résultats. Même si on dit souvent que Google est votre ami, il est toujours plus rapide et pratique d'être capable d'identifier rapidement la source précise du problème par nous-même.

Malheureusement il s'agit souvent de faire du debugging de code ou bien d'utiliser des outils qu'on ne connaît pas, donc rien d'évident à première vue.

Nombreux sont ceux qui ont passé des heures à debugger un code à la main sans comprendre précisément ce qui se passait derrière. Souvent parce qu'au delà du code et du binaire créé, existent des fonctionnalités du système que la documentation n'explique pas, mais qui interfèrent avec. L'idée de cet article est là : comprendre et connaître les méthodes employées par des programmes malicieux qui s'insèrent profondément dans le système afin de mieux les utiliser pour développer et debugger nos applications. Il s'agit aussi bien de techniques de code, que de manipulations de propriétés systèmes souvent cachées.

On peut aussi introduire la notion de sécurité, en effet savoir coder proprement c'est le faire de manière simple et sécurisée. Il faut donc être conscient des risques potentiels que peuvent comporter notre application. Dans ce cas, connaître les différentes failles et techniques mises en place par les hackers pour les exploiter, pour

nous permettre d'éviter de reproduire ces erreurs. Bien évidemment, il s'agit là d'abord de best-practices que l'on trouve dans les livres sur la programmation orientée sécurité ; mais c'est aussi un travail de veille car les techniques de protections et d'exploitations des systèmes évoluent constamment.

RTFM !

Un des principes de base que l'on oublie trop souvent, est de lire la doc ! On ne le dira jamais assez. Mais avant de commencer à émettre des hypothèses sur le mauvais fonctionnement du système il faut d'abord remettre en cause ce que l'on a fait.

Cela implique de bien comprendre ce qu'on a écrit et parfois, ce n'est pas évident, et cela parce que l'on fait appel à des fonctions extérieures. Dès qu'intervient l'utilisation de fonctions extérieures il faut tenir compte de la doc. Sous Windows, il existe la MSDN [1] qui est très bien faite. Prenons un exemple avec le code suivant :

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

#define PROCESS_NAME L"C:\\WINDOWS\\NOTEPAD.EXE"
int _tmain(int argc, _TCHAR* argv[])
{
    PROCESS_INFORMATION processInformation = {0};
    STARTUPINFO startupInfo = {0};
    DWORD status;

    if (GetFileAttributesW(PROCESS_NAME) == INVALID_FILE_ATTRIBUTES)
    {
        wprintf(L"Can't find " PROCESS_NAME L"; error %d, aborting\n", GetLastError());
        return(1);
    }

    startupInfo.cb = sizeof(startupInfo);

    if (!CreateProcessW(NULL, PROCESS_NAME, NULL, NULL, FALSE,
        0, NULL, NULL, &startupInfo, &processInformation))
    {
        status = GetLastError();
    }

    wprintf(L"Status launching " PROCESS_NAME L" is %d\n", status);

    return 0;
}
```

A priori ce code lance le binaire "notepad.exe". Pour cela, il vérifie que le fichier exécutable existe bien à l'aide de l'API `GetFileAttributesW` [2] puis lance le processus avec `CreateProcessW` [3]. Au moment de lancer le programme, celui-ci plante. La question est donc : pourquoi ?

Rien ne laisse a priori penser qu'il existe un dysfonctionnement dans ce code. On ne manipule que 2 structures qui sont initialisées correctement, il y a peu de chances que le bug provienne de la fonction `GetFileAttributesW` car le path du binaire n'est pas mal formé. On devrait donc remettre en cause la façon dont on utilise l'API `CreateProcessW` ? A cause de son grand nombre d'arguments il est facile de se tromper. Regardons son prototype :

```
BOOL WINAPI CreateProcess(
    __in_opt LPCTSTR lpApplicationName,
    __inout_opt LPCTSTR lpCommandLine,
    __in_opt LPSECURITY_ATTRIBUTES lpProcessAttributes,
    __in_opt LPSECURITY_ATTRIBUTES lpThreadAttributes,
    __in BOOL bInheritHandles,
    __in DWORD dwCreationFlags,
    __in_opt LPVOID lpEnvironment,
    __in_opt LPCTSTR lpCurrentDirectory,
    __in LPSTARTUPINFO lpStartupInfo,
    __out LPPROCESS_INFORMATION lpProcessInformation
);
```

En fait nous n'utilisons réellement que 3 arguments qui sont `lpCommandLine`, `lpStartupInfo` et `lpProcessInformation`. L'un d'entre eux provoque une erreur dans `CreateProcessW`. En regardant de plus près la documentation de cette fonction on lit :

lpCommandLine :

The command line to be executed. The maximum length of this string is 32K characters. If `lpApplicationName` is NULL, the module name portion of `lpCommandLine` is limited to MAX_PATH characters.

The Unicode version of this function, `CreateProcessW`, can modify the contents of this string. Therefore, this parameter cannot be a pointer to read-only memory (such as a const variable or a literal string). If this parameter is a constant string, the function may cause an access violation.

On apprend donc que `CreateProcessW` peut modifier le buffer qui contient `lpCommandLine`, or nous avons déclaré `PROCESS_NAME` en temps que constante à cause du `#define`. La fonction crée ainsi une exception que notre code n'est pas capable de gérer. Avec cet exemple nous avons pu voir qu'il est très important de lire complètement la doc.

Quand la documentation ne suffit pas, il faut passer par Google. Cependant il faut être prudent, privilégier les sites des références, c'est-à-dire les communautés de développeurs, leurs blogs, leurs mail-lists. Pour affiner ses recherches, il est intéressant d'apprendre quelques techniques de Google Hacking [4].

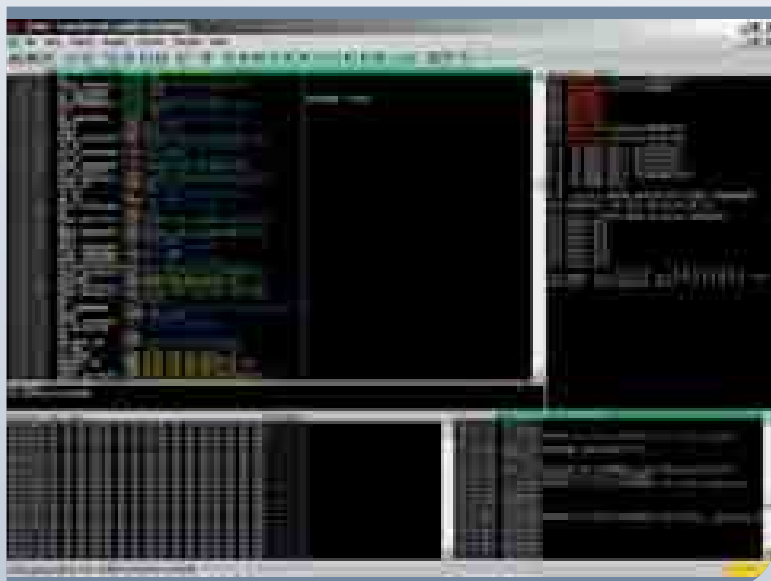
LA BOÎTE À OUTILS

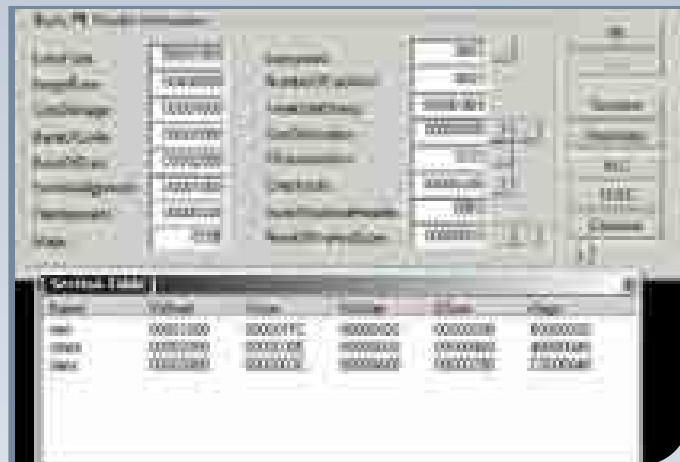
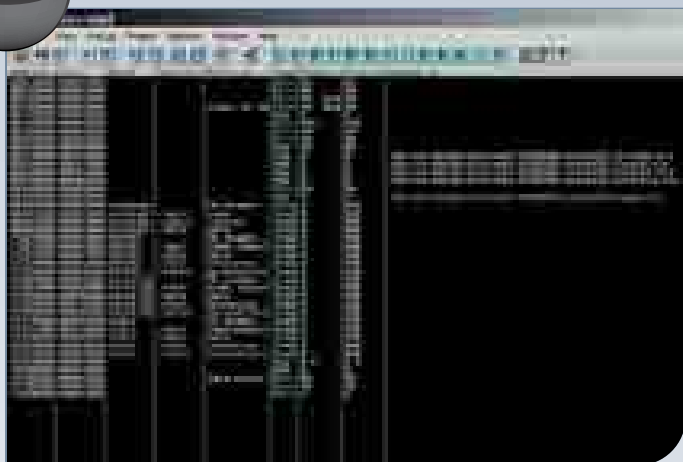
La plus grosse difficulté est de savoir quels sont les outils à utiliser et surtout comment ! On considère que nous n'avons pas lu la documentation relative à la fonction `CreateProcessW`. Notre premier outil sera OllyDbg [5], après l'installation, on le configure pour qu'il prenne la main lorsqu'une exception est générée. Il faut l'activer en faisant " Options -> Just-in-time debugging -> Make OllyDbg just-in-time debugger". Ensuite, lorsqu'un programme lance une exception la boîte d'erreur contient un bouton " Débogage " qui permet d'attacher OllyDbg sur le processus qui a planté pour étudier son état. Sur l'exemple plus haut, cela donne l'état suivant : [Fig.1]

On voit que c'est l'instruction " mov word ptr ds:[eax], bx " de `kernel32.dll` qui est la cause de l'erreur. En utilisant un outil comme Process Explorer [6], qui permet d'avoir des informations avancées sur les processus, on peut voir la stack du thread qui a généré l'exception. [Fig.2]

Pour l'instant, on sait juste que l'erreur provient de la fonction `CreateProcessInternalW` appelée par `CreateProcessW`. En regardant de plus près l'instruction fautive on s'aperçoit qu'il s'agit d'une écriture d'un word de valeur 0 à l'adresse `0x40214C`. L'état de la mémoire du processus est le suivant : [Fig.3]

L'adresse `0x40214C` correspond à la section ".rdata" du binaire. Avec l'aide de LordPE [7] on peut voir que les caractéristiques de cette section valent `0x40000040` (`IMAGE_SCN_MEM_READ` (`0x40000000`) | `IMAGE_SCN_CNT_INITIALIZED_DATA` (`0x00000040`)), ce qui correspond à une section en lecture seule avec des données initialisées. [Fig.4]





Notre code tente donc d'écrire dans cette zone mémoire ce qui entraîne une exception. Il suffit donc de changer notre déclaration en créant une variable globale de la forme " WCHAR PROCESS_NAME[]=L"C:\\WINDOWS\\NOTEPAD.EXE"; " et le tour est joué. La variable est créée dans la section " .data " du binaire qui est inscriptible.

Pour résumer, nous avons utilisé OllyDbg afin de trouver la source et le type d'erreur, avec Process Explorer nous avons affiné la localisation en affichant la pile d'appel du thread. On a vu que l'erreur provenait d'une tentative d'écriture dans une zone mémoire en lecture seule, ce que nous avons pu confirmer avec LordPE. Il nous a suffi ensuite de modifier notre code pour que tout fonctionne normalement. Évidemment, il ne suffit pas de savoir uniquement manipuler ces outils. Il faut aussi des connaissances de base sur le fonctionnement des processus et des threads ainsi que la manière dont sont utilisées les librairies et le format PE.

Il existe bien sûr d'autres outils, ceux de Sysinternals [8] proposent une vaste gamme d'utilitaires très bien faits comme Process Explorer, Process Monitor qui est une sorte de trace pour Windows ou Autoruns qui permet de voir ce que lance Windows au démarrage. Les Debugging Tools for Windows [9] sont aussi très puissants, ils fournissent notamment le debugger WinDbg. Enfin, on retrouve une bonne collection d'outils et de librairies sur la Collaborative RCE Tool Library [10].

HOOKING

Le hooking est une technique souvent utilisée par les malwares pour modifier le comportement d'une fonction. Il existe 2 types de hooking. Hooking par modification de la table des importations : La table des imports d'un binaire contient des pointeurs sur les fonctions externes qui sont situées dans des librairies. En modifiant ces pointeurs vers notre code on peut contrôler les appels du binaire vers ces fonctions. Cependant, il est impossible de hooker des fonctions internes au binaire avec cette technique.

Inline hook : Consiste à modifier le prologue d'une fonction avec un " trampoline " vers notre code, souvent il s'agit de remplacer les 5 premiers bytes par un " jmp <notre fonction> ". Cette méthode permet de hooker n'importe quelle fonction.

Il est possible sous Windows de transformer les hooks en LD_PRELOAD avec l'aide des Applnits_DLLs [11]. Il s'agit d'une clé de registre située dans HKEY_LOCAL_MACHINE\\Software\\Microsoft\\Windows NT\\CurrentVersion\\Windows qui contient un path sur une ou plusieurs DLLs qui seront chargées dans le processus au

moment de son lancement. Le code de la DLL, une fois chargé par notre processus, peut ensuite hooker les fonctions que l'on désire. C'est un hack souvent utilisé par des malwares pour s'injecter dans tous les processus de l'utilisateur. Voici un court code qui utilise la librairie Detours [12] pour hooker la fonction GetCurrentThreadId() et renvoyer tout le temps 1 à la place :

```
#include <windows.h>
#include <strsafe.h>
#include "detours.h"
#pragma comment (lib, "strsafe.lib")
#pragma comment (lib, "detours.lib")
DWORD WINAPI MyGetCurrentThreadId(void)
{
    return 1;
}
BOOL APIENTRY DllMain(HINSTANCE hInstance, DWORD dwReason, LPVOID lpReserved)
{
    CHAR Buff[512];
    static FARPROC RealGetCurrentThreadId;

    if(dwReason==DLL_PROCESS_ATTACH)
    {
        RealGetCurrentThreadId=NULL;

        StringCbPrintf(Buff, sizeof(Buff)-sizeof(char), "Dll injected
in process %lu\\n", GetCurrentProcessId());
        OutputDebugString(Buff);

        // Hook
        DetourTransactionBegin();
        DetourUpdateThread(GetCurrentThread());
        DetourAttach((PVOID)&RealGetCurrentThreadId, MyGetCurrentThreadId);
        DetourTransactionCommit();
    }

    if(dwReason==DLL_PROCESS_DETACH)
    {
        // Only if the hook is installed
        if(RealGetCurrentThreadId)
```



```

{
    DetourTransactionBegin();
    DetourUpdateThread(GetCurrentThread());
    DetourDetach((PVOID)&RealGetCurrentThreadId, MyGetCurrentThreadId);
    DetourTransactionCommit();
}
}
return TRUE;
}

```

Et voilà ! On a codé une DLL de hook, qui, couplée avec les `Applnits_DLLs` forme un `LD_PRELOAD` pour Windows, toujours pratique quand on veut debugger ou modifier du code.

MACHINES VIRTUELLES

L'utilisation de machines virtuelles (VM) est très pratique lorsqu'on développe des programmes qui modifient des composants systèmes. Il est impensable de coder et de tester un driver sur la même machine car si le driver plante, tout le système aussi. L'un des énormes avantages avec les machines virtuelles, c'est qu'on peut les "snapshot", c'est-à-dire sauvegarder un état de la VM et le restaurer. Ainsi si un de vos codes modifie énormément votre OS, vous évitez de l'abimer. Il existe plusieurs logiciels de virtualisation, les plus connus sous Windows sont VMware Workstation [13] et Virtual PC 2007 [14].

SÉCURITÉ

Enfin, il est intéressant d'être conscient des différentes protections fournies par le compilateur et l'OS [15]. Sous Visual C++ on retrouve :

- Le cookie de protection de pile (option /GS) qui sert à éviter les attaques de type stack buffer overflow.
- La vérification des handlers d'exceptions avec /SafeSEH. Les handlers d'exceptions sont positionnés dans la pile, si un attaquant est capable d'en contrôler un, il peut en générant une exception l'appeler et ainsi exécuter du code malveillant. L'idée du /SafeSEH est de fournir un mécanisme de vérification de ces handlers, prouvant que ce sont bien les handlers mis en place par le développeur. Sous Windows Server 2008 et Vista SP1 on trouve une nouvelle technique, le SEHOP.

Sachant que les handlers d'exceptions sont en fait une liste chaînée dans la pile du thread, si l'un d'entre eux est corrompu, la liste le sera elle aussi. Le /SafeSEH fournit une liste de ces handlers dans le binaire mais cela demande de le recompiler. L'idée, pour réaliser cette vérification dynamiquement est donc de mettre une valeur symbolique à la fin de la liste et de la parcourir à chaque exception afin de vérifier son intégrité.

Côté système on a :

- Le DEP (Data Execution Prevention) qui sur les architectures le supportant fournit une protection en exécution des pages mémoires. Par exemple, la pile ne devient plus exécutable, les attaques sont donc plus compliquées à mettre en œuvre, mais pas impossibles :]
- L'ASLR (Address Space Layout Randomization) qui existe depuis Vista. Afin d'empêcher un attaquant d'utiliser des morceaux de code qui sont situés à des adresses statiques en mémoire, le système randomize les adresses de chargement des modules.

CONCLUSION

Cet article avait pour but de présenter différentes techniques non traditionnelles pour aider au développement de code. Bien sûr, il en existe beaucoup plus qui peuvent être utiles à connaître, quelques recherches sur le net vous le montreront. J'espère en tout cas vous avoir donné envie d'aller plus loin sur le sujet afin d'améliorer vos applications aussi bien en termes de performances que de sécurité. Je remercie Programmez ! pour m'avoir donné l'occasion d'écrire cet article !

RÉFÉRENCES

- [1] MSDN Microsoft Developer Network <http://msdn.microsoft.com/>
 - [2] CreateProcess <http://msdn.microsoft.com/en-us/library/ms682425.aspx>
 - [3] GetFileAttributes [http://msdn.microsoft.com/en-us/library/aa364944\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa364944(VS.85).aspx)
 - [4] Google Hacking Database <http://johnny.ihackstuff.com/ghdb.php>
 - [5] OllyDbg <http://www.ollydbg.de/>
 - [6] Process Explorer <http://technet.microsoft.com/en-us/sysinternals/bb896653.aspx>
 - [7] LordPE <http://ivanleOu.free.fr/repo/softs/LORD-PE.ZIP>
 - [8] Windows Sysinternals <http://technet.microsoft.com/en-us/sysinternals/default.aspx>
 - [9] Debugging Tools for Windows <http://www.microsoft.com/whdc/dev-tools/debugging/>
 - [10] Collaborative RCE Tool Library http://www.woodmann.com/collaborative/tools/index.php/Category:RCE_Tools
 - [11] Working with the `Applnits_DLLs` registry value <http://support.microsoft.com/?scid=kb%3Ben-us%3B197571&x=12&y=13>
 - [12] Detours <http://research.microsoft.com/en-us/projects/detours/>
 - [13] VMware Workstation <http://www.vmware.com/fr/products/ws/>
 - [14] Virtual PC 2007 <http://www.microsoft.com/downloads/details.aspx?displaylang=fr&FamilyID=04d26402-3199-48a3-afa2-2dc0b40a73b6>
 - [15] Protecting Your Code with Visual C++ Defenses <http://msdn.microsoft.com/en-us/magazine/cc337897.aspx>
- Preventing the Exploitation of Structured Exception Handler (SEH) Overwrites with SEHOP <http://blogs.technet.com/swil/archive/2009/02/02/preventing-the-exploitation-of-seh-overwrites-with-sehop.aspx>
- IvanleOu - ivanleOu@security-labs.org

glossaire

Format PE : Format utilisé par les binaires Microsoft.

DLL : Dynamic-Link Library, une librairie de code qui est chargée dynamiquement dans un processus afin de lui fournir de nouvelles fonctionnalités.

Hooking : Technique qui consiste à "crocheter" une fonction afin de la détourner pour modifier son comportement.

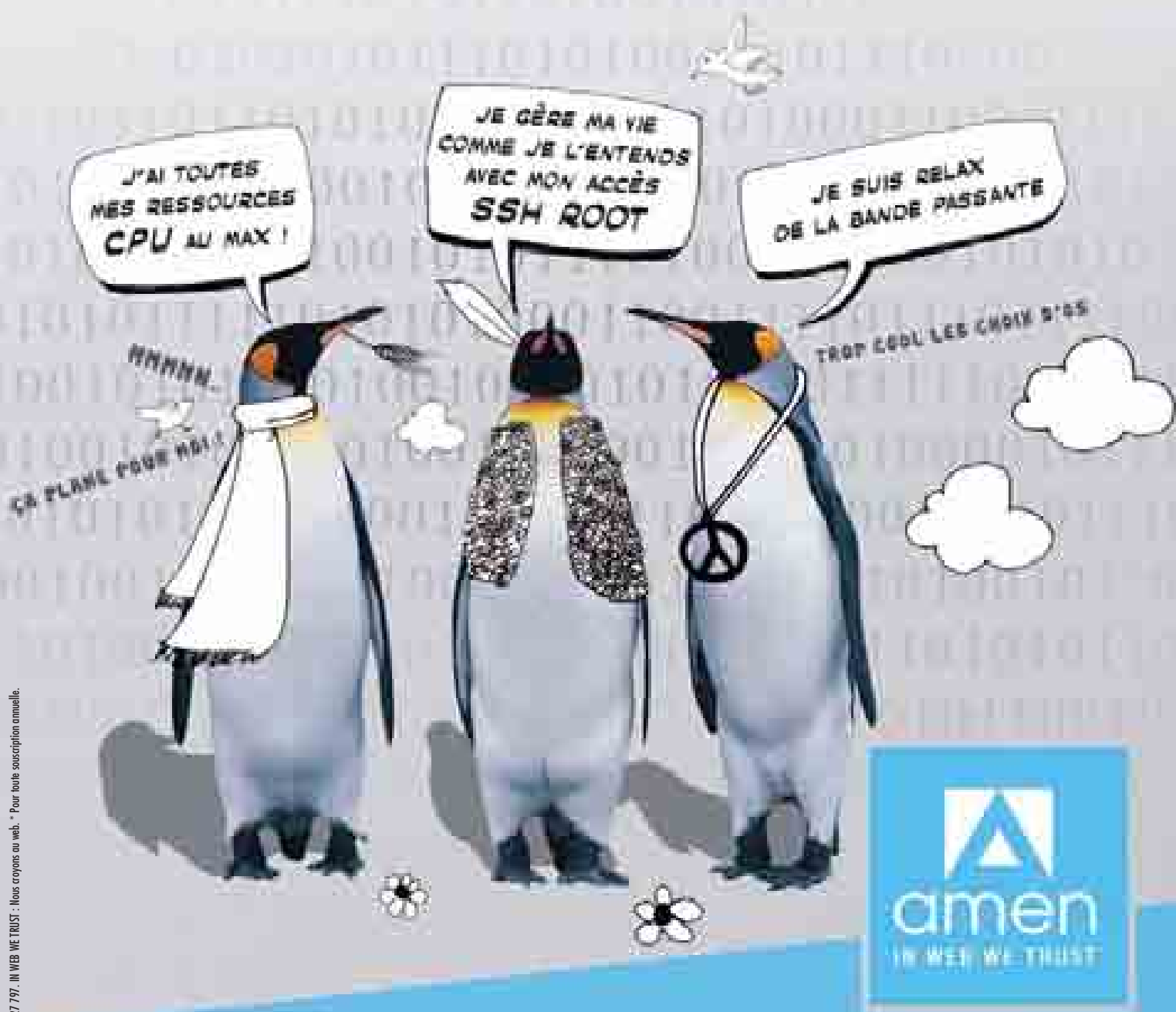
IAT : Import Adresse Table, table des imports d'un binaire qui contient les adresses des fonctions utilisées par le binaire et situées dans les DLL.

EAT : Export Adresse Table, table des exports d'une DLL qui permet au binaire de retrouver les fonctions pour les placer dans son IAT.

SEH : Structured Exception Handler, structure qui est placée dans la pile d'un thread et qui contient un pointeur sur le handler d'exception ainsi qu'un pointeur sur le SEH suivant.

Cookie : Valeur positionnée intelligemment dans la pile d'un thread afin d'empêcher d'écraser des variables locales d'une fonction en cas de stack overflow.

NOUVEAU VDS+ d'AMEN : le bonheur est dans le serveur !



À PARTIR DE
5€^{HT} /MOIS
soit 5,98 € TTC/MOIS*

**SERVEURS PRIVÉS AMEN :
BÉNÉFICIEZ DE RESSOURCES
GARANTIES QUI VOUS SONT
PROPRES (PROCESSEUR,
MÉMOIRE, DISQUE DUR...)
TOUT EN PROFITANT D'UNE
PLATEFORME INFOGÉRÉE
24H/24 - 7J/7.**

- Hébergement multi-sites/multi-domaines
- Interface d'administration : Plesk 8.6
- Systèmes d'exploitation : Fedora Core 8, Suse 10.3, Debian 4.0, Ubuntu 8.04 ou CentOS 5
- Part CPU minimum : de 1 à 6
- Mémoire garantie : de 256 Mo à 1 Go
- Espace disque : de 5 Go à 30 Go
- Bases de données : illimitées
- 1 adresse IP fixe
- Accès Root

**PARTENAIRE
INDUSTRIEL**



Pour plus de renseignements : 0892 55 66 77 (0.34 €/mn) ou www.amen.fr
NOMS DE DOMAINE - EMAIL - HÉBERGEMENT - CRÉATION DE SITE - E-COMMERCE - RÉFÉRENCIEMENT



La sécurité appliquée à .NET

La sécurité est un vaste sujet qu'il est toujours nécessaire d'aborder humblement étant donné la diversité, la complexité et la versatilité qui le caractérisent. Ce qui est connu un jour est remis en cause le lendemain par de nouvelles techniques, de nouvelles attaques et de nouvelles protections. Cet article a pour objectif de sensibiliser, sous divers angles d'approche, le lecteur développeur à quelques notions de sécurité qui touchent les développements .NET.

L'objectif d'intégrer la sécurité dans les développements applicatifs est d'offrir un degré de protection adapté en fonction du danger perceptible ou potentiel. Mais comment définir le danger ? Quel danger peut bien représenter une application ? En fait, il existe de nombreux procédés, que les personnes plutôt mal intentionnées utiliseront pour arriver à des fins pas très louables ! Parmi les plus connus : utilisation frauduleuse de données (bancaires ?!), destruction de données critiques, espionnage, vol d'identité, ... et la liste est tellement longue qu'elle pourrait faire un sujet d'article à part entière. Dans un monde où l'information devient une arme pour qui sait la manipuler, il est capital d'identifier les dangers et de prendre les mesures qui s'imposent dans nos développements afin d'éviter l'utilisation détournée de nos codes. Le "Security Development LifeCycle" (SDL) ou aussi appelé "Trustworthy Computing Security Development Lifecycle" est un processus décrit par Microsoft permettant d'intégrer les besoins relatifs à la sécurité tout au long du cycle de vie du développement d'une application. Ce processus est basé sur les bonnes pratiques de conception et de codage, la modélisation des menaces et les contre-mesures appropriées, la documentation et revues de code... Il existe des documents (référence n°3) et outils (référence n°2) permettant la modélisation et l'analyse des menaces comme, ci-dessous, le Microsoft Threat Analysis & Modeling v2.1.2. [Fig.1]

L'objectif de cette modélisation est d'identifier :

- **Les actifs de valeur**, tels que des données en base ou sur le système de fichiers (une table des clients en base),
- **Les menaces** potentielles d'endommagement des actifs (la perte de toutes mes données clients),
- **Les vulnérabilités ou faiblesses** qui font que les menaces peuvent être mises à exécution (l'accès en écriture à mes données clients),
- **Les attaques** qui représentent le mode de "passage à l'action" (injection du code SQL tel que TRUNCATE TABLE MesClients),

- **La contre-mesure** qui permet de minimiser un risque et de se préserver contre une menace (Un plan de backup régulier des données Clients ?). Ensuite, il convient d'identifier les menaces et de les catégoriser : **STRIDE**.
- **Spoofing** : consiste à utiliser une identité de manière frauduleuse (prendre l'identité de quelqu'un par exemple),
- **Tampering** : Modification ou altération non autorisée de données (exemple : modification de données dans une communication ou modification du binaire d'un programme),
- **Repudiation** : l'utilisateur doit pouvoir prouver qu'il n'a pas commis certaines actions (par exemple, le code CVC d'une carte bleue est utilisé par les banques pour prouver que c'est bien le possesseur de la carte qui l'a utilisée lors d'une transaction ; dans ce cas, le porteur de la carte ne peut répudier sa transaction, en théorie car la législation permet le recours),
- **Information disclosure** : C'est l'exposition d'informations à caractère confidentiel ou privé,
- **Denial of Service** : ou comment rendre un système ou une application inutilisables (comme par exemple une soudaine activité très forte sur un site Web qui rend le service inutilisable pour ses utilisateurs internautes),
- **Elevation of privilege** : Utilisation de privilèges plus élevés que ceux initialement accordés.

Les catégories de menaces nous aident à identifier des contre-mesures possibles qui nous permettront de protéger nos applications face à d'éventuelles attaques (le terme "éventuel" est choisi pour introduire ensuite la quantification du risque ou **DREAD**).

Le tableau ci-dessous donne une liste non exhaustive des contre-mesures applicables aux catégories de menaces :

Catégories de menaces	Contre-mesures possibles
Vol d'identité (Spoofing)	Utilisation d'une authentification forte. Ne pas stocker les secrets (mot de passe...) "en clair". Ne pas échanger de secrets "en clair". Protéger les cookies d'authentification avec SSL.
Altération de données (Tampering)	Utiliser la signature et le hash de données. Utiliser des protocoles de communication résistants face à l'altération de données et permettant de contrôler l'intégrité des messages échangés.
Répudiation	Avoir une traçabilité forte. Utiliser des signatures numériques.
Divulgaration d'information (Information disclosure)	Utiliser un chiffrement fort. Utiliser des protocoles de communication permettant d'assurer la confidentialité des messages échangés. Ne pas stocker les secrets "en clair" (comme les mots de passe)
Déni de service (Denial of service)	Utiliser des mécanismes de limitations de bande passante.
Élévation de privilèges	Suivre le principe de n'accorder que les privilèges nécessaires pour effectuer l'action demandée.



Fig.1

La sécurité étant un vaste et interminable sujet, les contre-mesures peuvent ne pas toujours résoudre l'ensemble du problème que pose une menace. Nous verrons d'ailleurs que .NET en résout certains par rapport au code natif, comme le buffer overrun (Rien n'est jamais acquis en sécurité), mais en apporte d'autres. Il est donc important de connaître le niveau de risque encouru face à une menace, car l'excès de sécurité pourrait nuire à vos développements, voire même être sa principale faille si les contre-mesures sont mal utilisées ou inadaptées : il faut éviter de tomber dans l'excès de paranoïa et **DREAD** est là pour ça !

La table ci-dessous propose un système de quantification permettant de calculer le risque face à une menace :

	Elevé (3)	Moyen (2)	Faible (1)
Dommage potentiel	L'attaquant peut tromper la sécurité, obtenir tous les accès, être administrateur, télécharger du contenu.	Fuite de données sensibles.	Fuite de données banales.
Reproductibilité	L'attaque peut être reproduite autant de fois que voulu.	L'attaque peut être reproduite à certaines conditions.	L'attaque est difficile à reproduire.
Exploitabilité	Un développeur novice peut exécuter l'attaque rapidement.	Un programmeur expérimenté pourrait exécuter l'attaque, puis répéter les étapes.	L'attaque ne peut être réalisée que par un développeur chevronné avec des connaissances pointues sur les couches basses.
Utilisateurs Affectés	Tous les utilisateurs.	Quelques utilisateurs.	Une très petite partie des utilisateurs.
Découverte	Des informations publiées expliquent l'attaque.	La vulnérabilité n'est connue que d'un groupe restreint d'utilisateurs.	Le problème est obscur et complexe et certainement difficile à découvrir.

Si nous appliquons cette mesure de risque à deux menaces très connues, nous obtenons les quantifications suivantes :

Menace	D	R	E	A	D	Total	Niveau
L'attaquant obtient des informations d'authentification via un outil de supervision du réseau	3	3	2	2	2	12	Elevé
Injection de commandes SQL	3	3	3	3	2	14	Elevé

Cette courte introduction à l'identification des menaces et contre-mesures pour vos développements étant faite, nous allons maintenant passer à des choses plus concrètes et des problématiques qui touchent les applications .NET.

Vol de propriété intellectuelle

Il n'y a pas très longtemps, je lisais sur le blog d'un développeur .NET qu'il avait retrouvé son application recompilée sur un site étranger avec une lettre de différence dans le nom. Son application étant disponible en téléchargement, il n'y avait rien de plus facile que de la rétro-concevoir étant donné qu'elle était compilée en MSIL, un langage intermédiaire permettant la portabilité sur différents systèmes exécutant .NET. Bien que l'application n'ait rien de particulier, c'est tout de même le fruit du travail de son auteur qui était en jeu. Imaginez seulement ce que cela peut donner sur un logiciel commercial ou issu de la recherche.

Comment décompiler un programme en .NET ? Rien de plus simple et tout bon développeur .NET s'y emploie chaque jour sur le framework pour parfaire ses connaissances avec des outils comme ILDASM ou Reflector qui est très largement utilisé. [Fig.2]

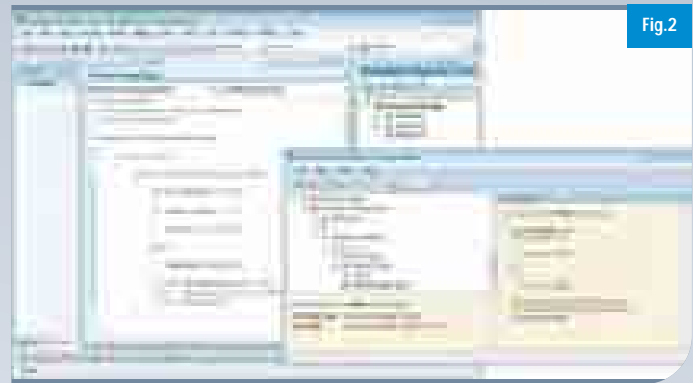


Fig.2

Offuscation

L'offuscation est l'art de brouiller les pistes ou encore obscurcir le code. Cette méthode ralentit considérablement la rétro-conception en fonction des algorithmes utilisés. Il existe environ une quarantaine d'algorithmes d'offuscation. Un algorithme d'offuscation possède en général 3 mesures permettant de connaître son efficacité : sa **puissance**, sa **robustesse** et son **coût**.

La **puissance** d'un offuscateur est la mesure selon laquelle le code est difficile à lire pour un humain. Par exemple, un des algorithmes d'offuscation est basé sur la destruction systématique de tout commentaire dans le code qui pourrait aider à comprendre son fonctionnement. Un autre algorithme d'offuscation consiste à modifier tous les noms des variables du code. Le programme fonctionne de la même manière mais est plus difficile à lire pour l'humain.

La **robustesse** d'un algorithme d'offuscation est la difficulté qu'aurait un dé-offuscateur (un programme) à régénérer une structure de code lisible. L'un des mécanismes les plus efficaces est celui de l'interprétation, mais c'est aussi l'un des plus coûteux : cela consiste à modifier le code MSIL pour le rendre interprétable par un runtime faisant partie de l'offuscateur.

Le **coût** représente par exemple la perte de performances à l'exécution induite par l'utilisation d'un algorithme d'offuscation. En général, plus l'algorithme est robuste, plus il est coûteux : il est nécessaire de bien connaître les effets des algorithmes d'offuscation sur l'exécution et sur la protection du code. Sans cette connaissance, le développeur pourrait se méprendre sur la qualité de la protection qu'il donne à son code.

L'exemple ci-dessous est tiré de l'utilitaire d'offuscation de Mpoware en version bêta (code source, puis code offusqué). Cet outil est le seul du marché qui permet d'offusquer du code source C# (offuscation pré-compilation) et non du MSIL (offuscation post-compilation) :

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

namespace Mpoware.CSharp.Parser.Tests
{
    public class Test5_ClassModifiers
    {
        // Avec un commentaire une ligne
        /*
        ** Small comment
        ** Delimited $$$
        */
    }
}
```



```
*/
public void Say(string message)
{
    i = 0;
    Console.WriteLine("Hello " + /*mon message*/ message);
}
}
```

Offusqué de la manière suivante :

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

namespace Mpoware.CSharp.Parser.Tests
{
    public class Test5_ClassModifiers
    {
        public void Say(string lCn)
        {
            int bFvkZBBQMMMybaxvs;
            bFvkZBBQMMMybaxvs = 0;
            if (bFvkZBBQMMMybaxvs == 0) goto sdfhsqdfmkjf;
            fmkjdsfhfkjf: if (bFvkZBBQMMMybaxvs == 1) { Console.WriteLine
            (GetString(new int[] { 2, 2, -3, 0, -1, 3 }) + lCn); goto end; }
            sdfhsqdfmkjf: ADDKSHFDDRH(bFvkZBBQMMMybaxvs); bFvkZBBQMMMybax
            vs++; if (bFvkZBBQMMMybaxvs == 1) { goto fmkjdsfhfkjf; } else {
            goto fmkjdsfhfkjf5; };
            bFvkZBBQMMMybaxvs = 0;
            if (bFvkZBBQMMMybaxvs == 0) goto sdfhsqdfmkjf;
            fmkjdsfhfkjf5: if (bFvkZBBQMMMybaxvs == 1) { Console.WriteLine
            (GetString(new int[] { 2, 2, -3, 0, -1, 3 }) + lCn); goto end; }
            sdfhsqdfmkjf6: ADDKSHFDDRH(bFvkZBBQMMMybaxvs); bFvkZBBQMMMybax
            vs++; goto fmkjdsfhfkjf;
            end1: ;
            bFvkZBBQMMMybaxvs = 0;
            if (bFvkZBBQMMMybaxvs == 0) goto sdfhsqdfmkjf;
            fmkjdsfhfkjf1: if (bFvkZBBQMMMybaxvs == 1) { Console.WriteLine
            (GetString(new int[] { 2, 2, -3, 0, -1, 3 }) + lCn); goto end; }
            sdfhsqdfmkjf2: ADDKSHFDDRH(bFvkZBBQMMMybaxvs); bFvkZBBQMMMybax
            vs++; goto fmkjdsfhfkjf;
            end2: ;
            bFvkZBBQMMMybaxvs = 0;
            if (bFvkZBBQMMMybaxvs == 0) goto sdfhsqdfmkjf;
            fmkjdsfhfkjf3: if (bFvkZBBQMMMybaxvs == 1) { Console.WriteLine
            (GetString(new int[] { 2, 2, -3, 0, -1, 3 }) + lCn); goto end; }
            sdfhsqdfmkjf4: ADDKSHFDDRH(bFvkZBBQMMMybaxvs); bFvkZBBQMMMybax
            vs++; goto fmkjdsfhfkjf;
            end3: ;

            end: return; ;
        }

        private int ADDKSHFDDRH(int bFvkZBBQMMMybaxvs)
        {
            return bFvkZBBQMMMybaxvs++;
        }
    }
}
```

```
private string GetString(int[] p)
{
    StringBuilder s = new StringBuilder();
    int[] al = new int[6] { 111, 108, 72, 32, 101, 34 };
    int i = 0;
    for (int k = 0; k < p.Length; k++)
    {
        i += p[k++]; s.Append((char)al[i]);
    }
    return s.ToString();
}
}
```

Peu de développeurs choisissent finalement de protéger leurs applications contre la rétro-conception par des offuscateurs, car il existe des inconvénients non négligeables et notamment les suivants :

- Brouillage relativement peu robuste pour un coût minimum,
- Certains offuscateurs peuvent altérer l'exécution de votre programme et le rendre inutilisable (cela peut arriver avec les applications utilisant la Reflection),
- Il existe des décompilateurs tels que Salamander .NET qui réduisent considérablement l'utilité de la plupart des offuscateurs,
- Il peut être plus difficile de débbugger un programme offusqué : la pile d'appels est elle-même difficile à lire par l'auteur du programme si l'offuscateur ne propose pas une table de lecture inverse...

Protection des assemblies par cryptage

Certains utilitaires comme .NET Reactor ou le Secure Vault de Mpoware permettent de chiffrer les assemblies à protéger ou une partie des ressources les plus critiques. Ci-dessous, nous vous proposons une technique simple que vous pouvez utiliser pour sécuriser vos assemblies. L'objectif est de chiffrer une assembly de calcul avec un code d'accès, de stocker l'assembly chiffrée dans les ressources d'une assembly appelante sous la forme d'un tableau de byte. Lors de l'utilisation de l'assembly de calcul, l'application demande le code d'accès et charge l'assembly calcul si le code d'accès est le bon.

1. Dans Visual Studio 2008, créez 2 projets : un Windows Forms et une librairie de classes. La librairie de classes contiendra vos classes de calcul.
2. Créez ensuite une application en ligne de commande permettant de chiffrer l'assembly librairie. Le code pourrait permettre d'entrer le path de l'assembly à chiffrer et un mot de passe, (**Voir le code source complet**).
3. Ajoutez l'assembly chiffrée dans les ressources de l'application .NET qui doit l'utiliser (Projet > Ajouter un élément existant). Modifier les propriétés du fichier " nomdufichier.Crypted.dll " et positionner " Build Action " à " Embedded Resource ".
4. Il ne reste plus qu'à implémenter le code de déchiffrement et le code d'utilisation de l'assembly protégée qui sera basé sur la Reflection (**voir le code source complet**).

Le cryptage d'assembly n'est pas à toute épreuve : il faut distribuer le mot de passe et l'assembly n'étant chiffré qu'avec un secret, tous les utilisateurs de l'application ont a fortiori le même mot de passe, à moins de compiler une version par utilisateur. De plus grâce à Reflector, le hacker peut lire la recette de déchiffrement de votre assembly. A ces problèmes, nous pouvons apporter plusieurs réponses pour renforcer la protection de notre propriété intellectuelle :

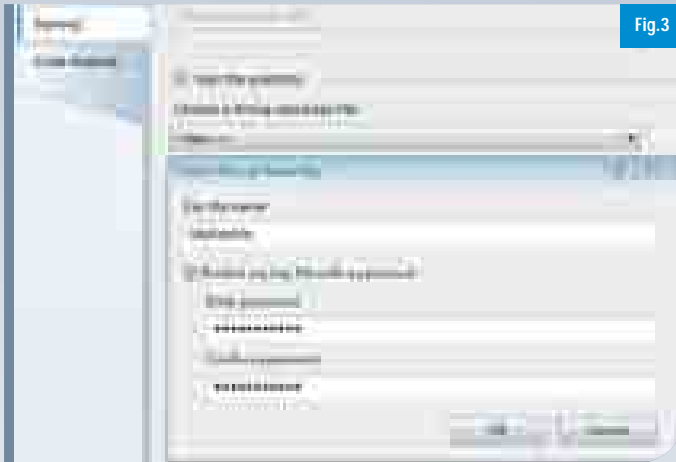


Fig.3

- Le secret peut-être une information connue de l'assembly appelante, afin de ne pas avoir besoin de demander un code à l'utilisateur,
 - L'assembly à charger peut vérifier l'identité de l'assembly appelante.
- Le nommage fort d'une assembly nous permet d'apporter des éléments de la solution : modifiez les propriétés de vos projets pour signer les assemblies. [Fig.3]

Pour que l'assembly appelée puisse vérifier l'identité de son appelant, nous modifions le code de " SecretClass " de la manière suivante :

```
public class SecretClass
{
    private const string GrantedCallingAssembly = "ACQAAASAA
ACUAAAABgIAAAkAABSU0ExAAQAAEAQBp6Voly5lr8Ev2EsqWE1V+oKssgU
QLnzf/uhBg4taWbfZN/PbYM+cPkp86DR+TCUhuszThwd1CSM9yKa5Xb0DQ5sH
m876Xz/6XhbjAdNor8MSy/IVy84BYz5wvFYI6kmbuSO049YwNiZFS5wx25rc/
WTwMIXUb4mFFkBAqmm5tA==";
    private bool _isGranted;

    public SecretClass()
    {
        _isGranted = false;
        // Récupération des informations de l'assembly appelante
        AssemblyName callingAsm = Assembly.GetEntryAssembly()
        .GetName();
        // Vérification de l'identité de l'assembly appelante
        if (Convert.ToBase64String(callingAsm.GetPublicKey())
        == GrantedCallingAssembly)
        {
            _isGranted = true;
        }

        public int Calculate(int value, int power)
        {
            // Vérifier si l'appelant est bien celui qu'il prétend
            être
            if (!_isGranted)
            {
                throw new SecurityException();
            }
            int res = 1;
            for (int i = 0; i < power; i++)
            {

```

```
                res = res * value;
            }
            return res;
        }
    }
}
```

Vous aurez à implémenter le code de récupération de la clé publique en base 64 pour écrire la constante GrantedCallingAssembly.

Conclusion

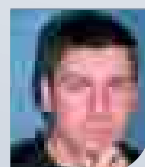
Nous avons introduit les notions de risques, niveaux de sécurité et réponses appropriées, puis, nous avons montré comment renforcer la sécurité contre le vol de propriété intellectuelle par l'obfuscation et le chiffrement. Lorsqu'il s'agit de protéger efficacement certaines parties de l'application, nous recommandons l'utilisation conjointe de l'obfuscation et du chiffrement. Etant donné qu'au bout du compte, le CLR doit tout de même exécuter le code, il n'existe pas de sécurité absolue puisque notre code est au moins lisible par le runtime. Le code natif C/C++ allié aux techniques d'obfuscation et de chiffrement est une bonne manière d'arrêter les outils de type Reflector et ILDASM. Le C++/CLI, permettant de mixer du code natif au code managé, peut-être une bonne solution pour encapsuler les parties sensibles de nos applications. Finalement, plutôt que de distribuer nos assemblies, ne serait-il pas plus judicieux de les déployer sur un serveur et d'en protéger l'accès ? Le modèle SaaS à base de services Web par exemple est un excellent rempart contre le vol de propriété intellectuelle et il n'impose qu'un seul point de sécurisation : la ferme de serveurs sur lesquelles sont déployées nos assemblies. C'est une alternative très intéressante au moment même où des plates-formes comme Windows Azure nous permettent de mettre en œuvre très facilement ce type de scénario. A très bientôt pour de nouvelles sécurités !

Références

1. Security Development LifeCycle : <http://msdn.microsoft.com/en-us/library/ms995349.aspx>
2. Microsoft Threat Analysis & Modeling v2.1.2: <http://www.microsoft.com/downloads/details.aspx?familyid=59888078-9DAF-4E96-B7D1-944703479451&displaylang=en>
3. Guide des bonnes pratiques "Improving Web Application Security: Threats and Countermeasures": <http://msdn.microsoft.com/en-us/library/ms994921.aspx>
4. Microsoft Application Threat Modeling Blog: <http://blogs.msdn.com/threatmodeling/>
5. Reflector: <http://www.red-gate.com/products/reflector/>

Glossaire

- **MSIL** : Microsoft Intermediate Language, c'est le langage assembleur exécuté par le runtime .NET.
- **Obfuscation** : pratiques permettant d'obscurcir du code, le rendre ambigu ou difficile à comprendre.
- **AppDomain** : Environnement isolé d'exécution d'applications .NET.



■ Frédéric QUEUDRET

MVP Client Application & CTO de la société Mpoware, accélérateur d'innovation. Société française d'édition de logiciels et de prestations de services sur les technologies .NET. <http://www.mpoware.com/>



Les 25 erreurs de programmation les plus fréquentes

Si vous n'intégrez pas la sécurité dans le cycle de développement de vos applications, je vous conseille alors de pratiquer l'art ancestral du Seppuku japonais : le résultat sera le même, quoique plus rapide, moins douloureux et n'affectera pas autant vos clients.

Pour les autres, moins friands de Shogun et plus de qualité, l'organisation SANS s'est associée à l'institut MITRE ainsi qu'à des groupes spécialisés dans la sécurité informatique (y compris le célèbre Homeland Security US, à travers sa division Sécurité Informatique), afin de nous apporter une synthèse des problèmes les plus fréquents (et des solutions possibles) rencontrés dans le développement des applications en matière de sécurité.

Quels en sont les destinataires : les développeurs bien évidemment, mais aussi les responsables du développement qui pourront s'assurer de la prise en compte des différentes vulnérabilités potentielles dans leur plan qualité. Et puis, oserais-je dire, tous les responsables à tous les niveaux, chez les clients comme chez les éditeurs, qui doivent comprendre les impacts dévastateurs possibles d'une absence de décision dans le domaine critique qu'est la sécurité. Afin d'atteindre cet objectif, il a fallu tout d'abord développer une méthodologie afin d'identifier les erreurs les plus importantes et les risques associés. Or,

ces erreurs ne sont pas actuellement suivies de façon très détaillée : si par exemple, les débordements de tampon (erreur la plus courante) font l'objet d'un suivi, cela n'est pas le cas pour les causes de ces débordements. Et les éditeurs de logiciels qui suivent ce type de problème à bas niveau, ne sont pas toujours prêts à partager leurs informations.



Critères d'importance

En l'absence de statistiques précises, ils ont choisi les critères d'importance suivants pour les erreurs, faiblesses et vulnérabilités sélectionnées :

- Élevée : la faiblesse a 50% de probabilité de se produire dans les logiciels potentiellement affectés
- Moyenne : la faiblesse a 10% à 50% de probabilité de se produire dans les logiciels potentiellement affectés
- Faible : la faiblesse a moins de 10% de probabilité de se produire dans les logiciels potentiellement affectés



Conséquences prévisibles

Que peut-il se passer si les vulnérabilités ne sont pas identifiées et corrigées ?

- Exécution de code : un attaquant peut exécuter du code ou des commandes, non prévus par le développeur
- Pertes de données : un attaquant peut voler, modifier ou corrompre des données sensibles
- Interdiction de service : un attaquant peut entraîner un arrêt ou un ralentissement du logiciel, et empêcher ainsi les utilisateurs légitimes d'accéder au logiciel
- Brèche de sécurité : un attaquant peut passer au travers d'un mécanisme de protection ; les conséquences dépendront de ce que le mécanisme était censé protéger.



Les conséquences des erreurs

Prévalence de la vulnérabilité	Faible/ Moyenne/ Élevée	Conséquences	Exécution de code Brèche de sécurité Perte de données Interdiction de service
Coût de réparation	Faible/Moyen/Élevé	Facilité de détection	Faible/Modérée/Élevée
Fréquence des attaques	Souvent/Quelquefois/ Rarement	Sensibilisation attaquant (est-ce une vulnérabilité connue ?)	Faible/Moyenne/Élevée

Afin de simplifier la lecture du rapport, une synthèse, sous forme d'un petit tableau récapitulatif, se trouve en tête de chaque rubrique. (Tableau page précédente)



Origines des faiblesses applicatives

Comme il n'est pas très facile de synthétiser un ensemble aussi varié, pourquoi ne pas nous mettre dans la peau d'un attaquant potentiel, ayant pris votre application en ligne de mire, et réfléchissant (sournoisement) à la meilleure stratégie d'attaque. Inutile de lire Clausewitz (ou plus récemment Petreaus) pour deviner que le premier angle d'attaque sera... vos écrans de saisie (ou pages web) ! Il commencera par mettre des valeurs non prévues dans TOUS les champs disponibles, et analysera le code JavaScript sous-jacent pour déterminer si, par le plus grand des hasards, vous n'auriez pas laissé la responsabilité de la sécurité à votre code client.

Puis il tentera de mettre tout et n'importe quoi dans les champs : ordres " <script ", commandes SQL, commandes au système d'exploitation, chaîne de caractères de longueur démesurée, etc. Si CHAQUE champ de saisie ne fait pas l'objet d'un contrôle strict répété sur le serveur, alors.....

La seconde vague sera sans doute lancée en coin ENTRE vos défenses, comme par exemple, en interceptant et remplaçant votre dialogue client/serveur par une version, disons " améliorée ". Ou en essayant de " débrancher " votre application ou site web vers une destination plus attractive pour l'attaquant (attaques XSS intersites). Transmettez-vous en clair des informations d'authentification, avez-vous oublié d'utiliser un protocole sécurisé (ou l'avez-vous développé vous-même, ce qui est la même chose) ? Aux galères !

Il se peut que vous donniez vous-même le mode d'emploi de l'attaque, en fournissant directement aux forces hostiles les plans de la forteresse : certains messages d'erreurs sont tellement détaillés qu'ils fournissent des informations très (beaucoup trop) précises, comme par exemple des chemins d'accès à des fichiers ou des listes de valeurs dangereuses.

Les utilisateurs apprécient les applications qui se mettent à jour automatiquement. Les forces des ténèbres aussi. Tout téléchargement de ressources, programmes, modules ou autres peut faire (FERA) l'objet d'une tentative de détournement. Le maître-mot dans cette situation s'appelle contrôle d'intégrité.

Si les murs du château semblent avoir une épaisseur suffisante, et si les échelles et tours de sièges sont repoussées avec succès, alors l'attaquant déterminé (et au moins aussi bien formé que vos équipes de développement) imitera l'inoubliable Albert Spaggiari, et passera...par les égouts.

Nous arrivons là à des attaques beaucoup plus sournoises et de beaucoup plus bas niveau. Elles demandent des connaissances exhaustives du fonctionnement inter-

ne des systèmes d'exploitation, ainsi que du cheminement interne d'une application. Malgré tout, nous en sommes toujours à un attaquant isolé, voire un petit groupe, et bien loin de ce que pourrait faire une organisation criminelle ou bien les spécialistes des services spéciaux d'un grand Etat (aucun lien ne pouvant bien sûr être fait entre ces deux types d'organisation, mais leurs approches techniques sont souvent similaires, alors que leurs buts et leur éthique diffèrent du tout au tout).

La protection des ressources sensibles, ainsi que la compétition naturelle entre différents processus ou modules de l'application pour leur contrôle peut être exploitée à des fins nuisibles. Si cette exploitation n'entraîne que l'arrêt imprévu du système, alors elle sera la base d'une interdiction de service. Au pire, elle induira le contrôle de l'application et de ses ressources par une force extérieure.

A ce niveau d'attaque on trouvera également les problèmes associés aux calculs arithmétiques (division par zéro et autres), l'utilisation de valeurs aléatoires (qui ne le seraient pas assez) et toute la problématique des générateurs de valeurs pseudo-aléatoires. Vous devez également prendre en compte le processus d'initialisation des variables de vos programmes. Il arrive quelquefois au compilateur de vouloir être plus intelligent que vous et de supprimer pour " optimiser " les fonctions d'initialisation. Un module pourrait donc démarrer avec des informations (sensibles) provenant d'une exécution précédente (faite peut-être avec un plus haut niveau de sécurité).



Attention aux accès

Pour finir cette vue générale de la psychologie criminelle, n'oubliez jamais que la résistance d'une chaîne est celle de son maillon

le plus faible : laisser la responsabilité d'assigner les privilèges d'exécution des modules, ou bien le niveau de protection des répertoires et fichiers au responsable surchargé de l'exploitation de l'application entraînera des déconvenues majeures. Si pour protéger votre application convenablement, il faut lire 20 pages de documentation, voire des appendices abscons dépendant des divers environnements possibles, alors vous pouvez dès à présent prévoir que cela ne sera pas fait.

La gestion des privilèges s'établit à la conception du logiciel, doit se vérifier module par module, dans tous les points d'entrée et TOUS les points de sortie, afin de les régler au niveau le plus bas nécessaire à chaque action. C'est absolument primordial pendant le processus d'installation et d'initialisation de l'application. Avez-vous utilisé par défaut les privilèges de l'installateur (surtout s'il s'agit d'un administrateur), avez-vous codé en dur un mot de passe ou compte utilisateur pour la première utilisation, avez-vous mis les fichiers temporaires dans une zone accessible au tout venant ? Aux galères, vous dis-je !

En conclusion, une remarque de bon sens : toutes ces erreurs sont généralement commises de bonne foi et seule l'intégration de la sécurité dans le cycle de développement (dès la conception) pourra diminuer (et je n'ai pas dit supprimer...) les vulnérabilités générées.

Sauf une. Il est impardonnable (et rarement pardonné) de mettre des informations de connexion en dur dans vos applications, car cela compromettra d'un coup la sécurité de TOUS vos clients.



Méthodes de prévention

Chacune des 25 faiblesses fait l'objet, dans le rapport, d'une analyse détaillée pour la prévention des erreurs ou l'atténuation de leurs conséquences. La présentation couvre les domaines suivants :

• Conception et architecture de l'application

Cela passe par la construction d'une liste la plus exhaustive possible des causes probables de la faiblesse, et pour chacune d'entre-elles, d'une ou plusieurs méthodes de prévention ou d'atténuation. On trouvera dans cette rubrique des suggestions de Framework de référence avec des conseils d'utilisation (et des limites de couverture des risques). Egalement des conseils, souvent de bon sens, quelquefois plus " pointus ", en matière de programmation, ou de codage des modules. Ces conseils sont organisés en fonction des langages potentiellement utilisés et de leurs dangers. On y verra

aussi des réflexions sur les différents protocoles de communication ou de codage de caractères et leurs incohérences, ainsi que des faiblesses structurelles de certains environnements ou systèmes. Une très bonne analyse des précautions à prendre dans l'utilisation des protocoles sécurisés et de la cryptographie fait partie de la panoplie proposée à l'architecte et au concepteur. On notera une analyse détaillée de certains problèmes courants rencontrés avec " C " et un rappel des interpréteurs modernes, censés pallier ces difficultés, mais qui sont tout de même...écrits en " C ".

• Implémentation

Nous rentrons là dans le détail des " trucs et astuces " et des précautions à prendre pour la mise en place des solutions. Cette section va jusqu'à la description détaillée de l'ordre des séquences de contrôle à mettre en place et des références constantes à des erreurs de plus bas niveau qui pourraient être générées par la correction de la faiblesse considérée. Chaque Framework est présenté dans le contexte de la faiblesse à pallier, et sera donc détaillé plusieurs fois dans le document en fonction de ses différentes utilisations. Une attention particulière a été apportée à l'analyse des " frontières ", ou espaces de transition entre différentes technologies hétérogènes. Cela couvre l'appel d'un langage à partir d'un autre au fonctionnement éloigné (code natif à partir de Java par

exemple), ou bien la communication entre plusieurs parties d'une même page web écrites avec des langages, voire des codages de caractères différents. Pour chacune des faiblesses, les conséquences des différents choix de procédures ou de fonctions sont détaillées, afin de ne jamais donner l'illusion d'une sécurité totale. Les auteurs n'hésitent pas à donner des exemples précis et détaillés.

• Opération ou environnement d'exploitation

Pour pallier certains problèmes potentiels, il est souvent nécessaire de mettre en œuvre des outils externes à l'application, comme un pare-feu par exemple. Ces outils sont détaillés au cas par cas avec, le cas échéant, leurs limites d'utilisation.

• Configuration système

Ce domaine couvre tout ce qui concerne la protection des ressources applicatives par le système d'exploitation. Cela comprend (mais n'est pas limité à) la protection des fichiers et répertoires, l'organisation et l'accès aux fichiers de configuration, aux fichiers temporaires ou aux privilèges à l'exécution des différents programmes et modules.

• Tests

Les tests nécessitent des procédures particulières en fonction de chaque vulnérabilité potentielle. Pour chacune on trouvera les outils standard, s'ils existent, ou des méthodes précises à mettre en place pour être le plus exhaustif possible dans la mise au point des solutions recommandées.

Ces méthodes couvrent l'utilisation de points d'arrêts, de " ralentisseurs " de processus ou toute autre technique appropriée.

• Compilation et Edition de liens

Il arrive que nos outils outrepassent leurs droits en croyant bien faire. Bien sûr, retirer l'option DEBUG lors de la génération de la version de production fait partie des contrôles de base, mais on oublie souvent que le code généré dans cette configuration sera différent de celui de la version de mise au point. Non seulement des procédures d'initialisation de variables pourraient être supprimées, mais également les bibliothèques utilisées par l'édition de liens pourraient ne pas avoir exactement le même comportement que celles utilisées durant la phase de validation.

Dans chaque cas, les paramètres à fournir aux compilateurs seront détaillés, ou au moins le comportement attendu sera précisé pour ceux non couverts par l'étude. Nous vous invitons à télécharger le rapport, en français, sur notre site www.programmez.com. Et n'hésitez pas à venir en discuter sur nos forums ! Et n'oubliez pas : il vaut mieux prévenir que guérir !

■ Gilbert Vidal

C.T.O. idSM, éditeur de logiciels basé à New York





Les réactions

// UN OUTIL POUR CHAQUE PROBLÈME



Bernard Ourghanlian

(directeur technique et sécurité, Microsoft)

Ce rapport remet un peu ce problème au goût du jour. Il y a quelques années, 90 % des vulnérabilités concernaient le système et 10 % les applications, aujourd'hui, c'est l'inverse. L'application pose problème. Ce document peut au moins diffuser quelques bonnes pratiques. Pour chaque problème il faut savoir quel outil utiliser. Il faut aussi modéliser la menace. Un code sécurisé revient 10 à 15 % plus cher qu'un code "normal". Trop souvent, la sécurité passe à la trappe (dans les projets). Il faut trouver les moyens pour améliorer le code actuel, sans pour autant compliquer le langage de programmation.

// ACCOMPAGNER UNE APPLICATION SUR TOUTE SA DURÉE



Arnaud Maillard

(directeur du laboratoire sécurité, Epitech)

Ce rapport, même s'il s'agit d'une initiative louable, s'inscrit toutefois dans une démarche problématique, elle-même sous-jacente de nombreux problèmes de sécurité. A l'heure actuelle, bon nombre d'applications (selon la maturité de leur cycle de développement) affichent des erreurs de programmation, et il est inévitable que celles-ci débouchent souvent sur des failles. Mais il s'agit là de phénomènes s'inscrivant dans l'orientation d'un projet, dans sa mauvaise conception d'ensemble. Recenser des erreurs, les corriger. Il s'agit là de notions qui doivent accompagner une application sur toute sa durée de vie. De fait, un rapport a peu de sens, dès lors que l'on considère la sécurité comme un domaine mouvant, dont les pierres d'achoppement aujourd'hui ne sont pas nécessairement celles d'hier. Poser les problèmes en termes clairs et précis pour pouvoir les résoudre, chacun en sécurité informatique y travaille. Mais scléroser un modèle, c'est aussi en masquer les zones d'ombre, les parties de développement où, justement, on ira sans doute découvrir les failles de demain.

// PERMETTRE À NOS CLIENTS DE REMÉDIER À CETTE PROBLÉMATIQUE



Olivier Jean

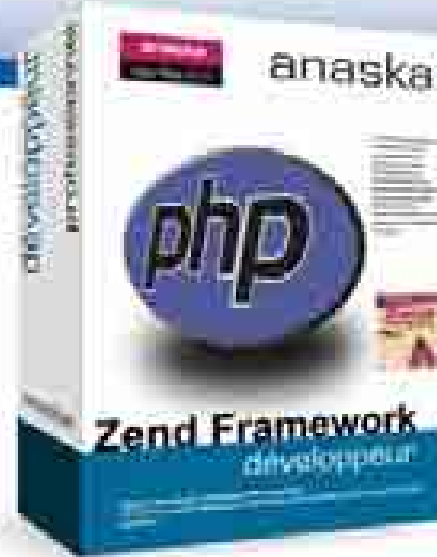
(consultant HP Software, expert des solutions de sécurité applicative)

Ces 25 erreurs sont assez récurrentes sans que rien ne soit fait (mis à part quelques bonnes pratiques distillées auprès des développeurs). Nous sommes tellement d'accord avec ce rapport que nous avons racheté SPI

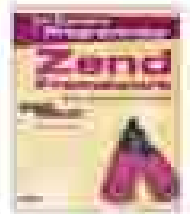
PRENEZ DE LA HAUTEUR

PHP > Framework

Formation au Zend Framework



"Une formation mise en place par l'auteur du livre Zend Framework"



Au Programme des 4 jours :

- * Introduction au Zend framework
- * Installation et configuration
- * La Programmation Orientée Objet
- * Composants du noyau
- * Gestion des bases de données
- * Modèle Vue Contrôleur (MVC)
- * Sécurisation et gestions de droits
- * Fonctionnalités avancées du Zend framework

Prochaines sessions

Paris 2009
19/01, 09/02, 09/03
06/04, 04/05, 02/06

Lyon 2009
09/02, 09/03, 06/04
04/05, 29/06

Tarif
1700 € HT

LE SPÉCIALISTE DE LA FORMATION POUR L'OPEN SOURCE

Informations

01 45 28 09 82

www.anaska.com



pour permettre à nos clients de mettre en place des processus et outils pouvant remédier à cette problématique. Il s'agit exactement du message que nous portons à nos prospects "inclure les tests de sécurité dans l'ALM et si possible le plus tôt".

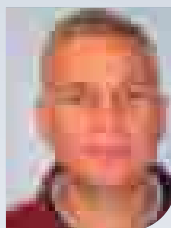
// CE RAPPORT ABORDE SURTOUT LES APPLICATIONS WEB



Vincent Untz
(développeur Gnome chez Novell)

D'une manière générale, les erreurs figurant dans le rapport sont assez fréquentes et dangereuses. C'est une bonne chose que des experts travaillent sur l'explication de ces erreurs et disent pourquoi il y a des conséquences réelles et pourquoi on doit s'en préoccuper. Il est réellement facile d'introduire, par mégarde, une faille de sécurité dans son code ! Une petite critique. Ce rapport aborde surtout les applications web. Ce serait une bonne idée d'avoir des rapports complémentaires ciblant les programmeurs web, les programmeurs d'applications systèmes etc. Malheureusement, je ne pense pas qu'un tel rapport soit suffisamment lu. Quand à l'implémentation (des solutions), je serai là aussi un peu négatif. Examiner du code et s'assurer qu'il n'y pas de problème de sécurité revient très cher. Beaucoup de développeurs pourraient écrire mieux, mais ils ne regardent pas en arrière, sauf si un défaut de sécurité est découvert. Une approche est la mise en œuvre de la sécurité dès le début du projet. Mais cela signifie qu'il faut prendre du temps pour mettre en place les fonctions et mécanismes de sécurité...

// POUR LES FRANÇAIS ET LES EUROPÉENS, IL N'EST PAS UNE CHOSE NOUVELLE



Eric Filiol
(directeur du laboratoire de virologie et de cryptologie, ESIERA)

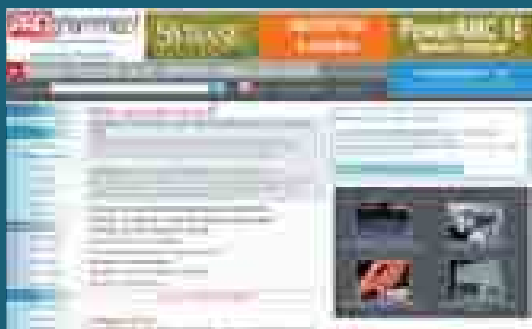
En fait les vulnérabilités logicielles et la programmation sécurisée sont présentées comme une problématique en apparence nouvelle, non sans un certain effet de mode. Cela fait près de 9 ans qu'en France, le problème est non seulement connu, mais également abordé sur le plan technique : entre les célèbres articles de Raynal/Blaess sur la programmation sécurisée et les cours dispensés aux élèves ingénieurs et masters SSI de l'ESIEA, je dirais que la France a une culture dans ce domaine qui fait défaut encore aux Anglo-saxons qui semblent ENFIN découvrir le problème. De fait, si un standard devait être instauré, je pense que l'école française de programmation serait la meilleure inspiratrice. Il est vrai que la culture française est plus tournée vers la qualité technologique que vers l'acte de commerce, culture éminemment anglo-saxonne qui abrite la plupart des grands éditeurs logiciels... et qui sous-traient la programmation en Inde ou dans des pays émergents. Or le problème de la sécurité tient au fait que le service est commercialement plus important (surtout dans un contexte de concurrence économique) que la sécurité qui elle, nécessite de prendre un peu plus de temps et de maîtriser les concepts. Donc à mon sens si ce rapport est intéressant, il n'est pas, au moins pour les français et les européens une chose nouvelle.



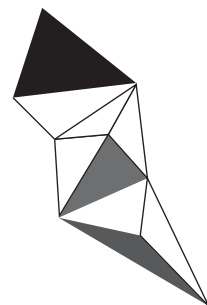
PROgrammez! Le magazine du développement

Les 25 erreurs
de programmation
les plus fréquentes

EXCLUSIF



Le rapport **complet**
en français
est disponible
sur
www.programmez.com



Devenez en 5 ans un expert en informatique reconnu

PARIS
BORDEAUX
LILLE
LYON
MARSEILLE
MONTPELLIER
NANCY
NANTES
NICE
RENNES
STRASBOURG
TOULOUSE



ADMISSION :
APRÈS BAC
(TERMINALES GÉNÉRALES OU STI),
BAC+2, BAC+3.

www.epitech.eu

01 44 08 00 50

European Institute of Information Technology - Titre homologué par l'Etat - Niveau I (CNCP)

Cette école est membre de **IONIS** ÉTABLISSEMENT D'ENSEIGNEMENT SUPÉRIEUR PRIVÉ - ASSOCIATION À BUT NON LUCRATIF (LOI 1901)

EDUCATION GROUP

Comment les éditeurs séduisent les étudiants et les écoles

Les éditeurs multiplient les actions envers les écoles et universités. Enseignants et étudiants y gagnent des conditions avantageuses pour acquérir et utiliser des logiciels. Quant aux éditeurs, ils constituent ainsi un vivier de compétences pour leurs propres besoins et pour ceux de leurs clients et partenaires.

Programme "Microsoft Students Partners": les étudiants deviennent ambassadeurs de Microsoft.



Les éditeurs de logiciels cherchent à se faire connaître des étudiants et à être présents dans les établissements d'enseignement. Objectif annoncé : donner aux étudiants une formation directement monnayable auprès des employeurs. En réalité, ils veulent surtout favoriser un réflexe auprès de la population des jeunes, afin que, une fois dans l'entreprise, ils choisissent de préférence les produits qu'ils connaissent et maîtrisent le mieux. Pour une somme symbolique par an, certains éditeurs, notamment Microsoft et Oracle, mettent à la disposition des écoles une grande partie de leur catalogue. Selon les partenariats, les éditeurs peuvent mettre à disposition des établissements ou des étudiants divers avantages : cours (e-learning), téléchargements, support, organisation d'événements, blogs, nouvelles et

actualités, forums, foires aux questions, etc. Sans surprise, Microsoft est largement en tête en nombre de partenariats et en multiplicité des actions menées vis-à-vis des étudiants, des écoles et des universités.

Microsoft omniprésent

" Nous avons depuis bientôt 10 ans des partenariats avec l'enseignement supérieur et des tarifs préférentiels dans l'éducation ", rappelle **Laurent Ellerbach**, marketing division plateformes et écosystèmes, responsable de la politique et des actions de Microsoft vis-à-vis de l'enseignement, qui fait état de 700 établissements partenaires en France. Parmi ces actions, la MSDN Academic Alliance a pour fonction de fournir les outils aux étudiants d'une école, et couvre la quasi-totalité des logiciels Microsoft. Un abonnement forfaitaire

annuel payé par l'établissement permet l'utilisation de ces logiciels à l'école et à domicile par les étudiants. Citons aussi Imagine Cup, le concours international organisé chaque année (Programmez! n°111). " Par ailleurs, les étudiants peuvent télécharger des logiciels pour une utilisation à but non lucratif (programme DreamSpark). 350 000 logiciels ont ainsi été téléchargés l'année dernière ", ajoute Laurent Ellerbach.

Autre action marquante, Microsoft met en place dans les établissements d'enseignement des " Student Partners " (MSP). Ils sont plus de 2000 étudiants dans plus de 100 pays à travers le monde, et une centaine en France. Basé sur le bénévolat, ce rôle, assuré par des étudiants, consiste à relayer l'information concernant les nouveaux produits et technologies Microsoft auprès des



Laurent Ellerbach, Microsoft

autres étudiants, et éventuellement des lycéens. Ils gèrent aussi le programme de téléchargement gratuit des outils dans leur campus. " Nous leur faisons découvrir les nouvelles technologies, nous faisons des conférences et des animations dans les écoles ", explique un MSP de Paris VII. Condition : être passionné par les technologies avancées. (cf. encadré)

Oracle très actif dans le monde étudiant

Plus de 80 établissements en France sont adhérents au programme Oracle Academy. Les avantages : des tarifs " symboliques " pour des logiciels et matériels pédagogiques, l'accès à des " briques " pour construire les cours. " Tous les établissements d'enseignement qui délivrent des diplômes peuvent accéder à cette offre ", explique **Dominique Van Deth**, responsable des programmes éducation, Oracle France, qui explique l'intérêt de ces partenariats : " Une logique de recrutement de jeunes diplômés pour Oracle et son écosystème (SSII). Par exemple, nous avons un partenariat avec l'Ecole de management des systèmes d'information



Dominique Van Deth

(EMSI) de Grenoble pour nos besoins propres en tant qu'éditeur de logiciels. " Oracle Academy permet aussi de préparer à la certification : " Les étudiants ont une remise importante sur le coût de la certification pour les écoles faisant partie du partenariat ", commente Dominique Van Deth.

Par ailleurs, Oracle propose aussi des avantages au niveau lycée ou même avec des classes de Troisième, via Oracle Education Foundation. Cette ONG fondée par Oracle propose dans le monde entier ThinkQuest, une plate-forme d'apprentissage en ligne pour les écoles primaires et secondaires. Elle permet aux enseignants et aux élèves de mettre en œuvre un apprentissage collaboratif. " Des écoles peuvent utiliser gratuitement la plate-forme collaborative ThinkQuest, hébergée par Oracle. Grâce à cette plate-forme, qui permet aux écoles de communiquer au format numérique, les élèves peuvent se familiariser avec la collaboration, les blogs, la sécurité ", ajoute Dominique Van Deth.

Les applicatifs et l'enseignement

Aux logiciels de base pour les programmeurs, les éditeurs ajoutent des applications de type ERP ou CRM. C'est le cas du programme " Enterpri-

se Business Application " d'Oracle Academy, qui intègre E-business Suite, Enterprise, Edwards EnterpriseOne, Siebel CRM, ainsi que des technologies de Business Intelligence et applications de gestion de la performance financière. Des éditeurs d'applications techniques ont aussi des programmes pour l'enseignement. A titre d'exemple, dès 2000, l'académie de Montpellier a souhaité équiper ses filières technologiques d'une chaîne numérique complète de CFAO (conception et fabrication assistées par ordinateur). Le logiciel Autodesk Inventor Professional avait alors été retenu pour l'ensemble des filières technologiques. Montpellier a récemment renouvelé son partenariat en s'équipant de 940 licences Autodesk Inventor 2009, grâce au programme Autodesk Education. Par ailleurs, Autodesk met à disposition un portail documentaire pour les étudiants en conception et en ingénierie. Il offre aux étudiants la possibilité de télécharger gratuitement des versions étudiantes complètes des logiciels et des ressources d'apprentissage. Plus de 500 000 licences gratuites ont été téléchargées par des étudiants et enseignants du monde entier. Il s'agit de solutions 2D et 3D pour l'enseignement et l'apprentissage de la conception dans les

Un exemple de partenariat école-éditeurs : Supinfo

Supinfo a une ancienne culture du partenariat avec les éditeurs informatiques. Ce sont même ces partenariats qui ont " sauvé " l'école lorsqu'elle était en danger de disparaître. " En 1999, Microsoft nous a présenté son programme 'Compétences 2000' et nous avons démarré avec eux pour former les enseignants et préparer des certifications Microsoft ", raconte Alick Mouriesse, président de Supinfo. A l'origine, il s'agissait, pour l'école, d'avoir des enseignements à jour sur ces technologies très demandées par les entreprises. Par la suite, plusieurs " laboratoires pédagogiques " ont été mis en place, dont chacun est consacré à une technologie ou un éditeur : Microsoft, Cisco, .Net, Apple, Sun, Oracle, Linux, IBM, Mandriva. " Nous considérons les partenariats avec des éditeurs de logiciels comme très importants pour la formation à ces produits ", explique Mauricio Diaz Orlich, directeur des laboratoires Supinfo. " Nous n'avons pas la même relation avec tous les partenaires. Par exemple, Microsoft nous permet d'accéder à ses supports de cours, et nous avons prati-



quement ligne ouverte pour poser nos questions ; par ailleurs, Supinfo est aussi la première ressource française pour Microsoft, nous sommes un site de référence. Pour Cisco, nous bénéficions du support. Oracle a mis à notre disposition sa base de données et nous avons un accès privilégié aux supports. Il y a une collaboration avec les constructeurs ou les éditeurs pour mettre à la disposition des étudiants des supports de cours et les préparer aux certifications. Enfin, des anciens élèves travaillent chez les partenaires, ce qui favorise aussi les bonnes relations. "

domaines de la fabrication, de la conception industrielle, de l'architecture, de la construction, du génie civil, des médias et du divertissement. " Nous proposons un tarif spécifique éducation, avec réduction de 98% par rapport au tarif entreprises ", précisent Corinne Cordeiro, responsable des ventes Education France, et Marie-Anne Bloch-Domont, directeur Channel & Sales Operations France, Autodesk. Pour les établissements partenaires, les logiciels Autodesk Education sont offerts pour un an aux professeurs et aux étudiants qui peuvent les télécharger gratuitement. " Sur le portail étudiant d'Autodesk, il y a eu plus de 20 000 téléchargements de logiciels depuis 2 ans ", indique Corinne Cordeiro. De plus, des formations gratuites sont dispensées aux enseignants chez Autodesk. " L'éducation est un pilier stratégique pour le développement d'Autodesk en France et en Europe du Sud. "

Les étudiants et les formateurs fran-

çais peuvent découvrir les avantages de la communauté Autodesk des étudiants en conception et ingénierie dès aujourd'hui en s'inscrivant depuis le site du Micro Portable Etudiant proposé par le Ministère de l'Education Nationale et le Ministère de l'Enseignement Supérieur et de la recherche. " La communauté en ligne donne accès aux logiciels gratuits d'Autodesk et la possibilité aux étudiants de télécharger gratuitement des logiciels professionnels ", déclare Roland Cernat, lauréat du concours Student Design Challenge dans la catégorie Conception industrielle et du Grand Prix du Développement Durable. " Le portail étudiant propose une large gamme de programmes, ce qui m'a permis d'essayer et de choisir l'outil le plus adapté à mes besoins. Etudiant en conception industrielle, j'utilise particulièrement le logiciel Autodesk AliasStudio. Pendant l'année scolaire, la communauté était le lieu principal où je venais chercher de

l'aide ou me tenir informé sur l'actualité des entreprises. La communauté m'apporte les outils et les conseils d'experts, nécessaires pour mes études comme pour me préparer à ma future activité professionnelle. "

Des programmes institutionnels

Le Micro Portable Etudiant, proposé par le Ministère de l'Education Nationale et le Ministère de l'Enseignement Supérieur et de la recherche, fait partie des Mesures Incitatives pour les Etablissements (MIPE) visant à doter ceux-ci d'infrastructures (Wifi), de services et d'accès aux ressources logicielles, éditoriales, pédagogiques. Plusieurs éditeurs et constructeurs informatiques sont partenaires de ce programme, dont Microsoft, Intel, HP, Dell, Apple, Autodesk, 4D, Adobe, etc. Par ailleurs, il existe des coopérations pour la formation fondées sur des partenariats entre des éditeurs et une région. A titre d'exemple de création de coopération réussie pour la formation des éditeurs en Rhône-Alpes : Syntec informatique et le Cluster Edit (espace fédérateur permettant la mise en relation les acteurs de l'édition du logiciel) ont mis en place trois nouvelles actions collectives de formation spécialement conçues pour répondre aux besoins des éditeurs de la région Rhône-Alpes. 300 salariés devraient bénéficier de ces actions de formation en 2009.

■ Claire Rémy

Le témoignage d'un MSP

(Microsoft Student Partner) :
Anthony Sharpe,
étudiant à l'Ecole supérieure
d'ingénieurs Léonard de Vinci (ESILV)

Qu'est-ce qui vous a amené à être MSP ?

Je voulais intégrer un club d'informatique dans mon école. En essayant de m'informer sur le site Microsoft, j'ai découvert le programme MSP. Pour devenir MSP, il faut envoyer sa candidature avec lettre de motivation. Il n'est pas nécessaire d'être étudiant en informatique, on peut aussi être dans une école de commerce, par exemple.

Quels contacts avez-vous avec Microsoft et d'autres MSP ?

J'ai accès à des forums de discussion avec d'autres MSP. Je reçois régulièrement des mails pour annoncer des événements, des nouveautés, des meetings organisés par Microsoft. Par exemple, en octobre dernier, j'ai participé aux MSP Days dans les locaux de Microsoft. J'ai des liens avec les équipes de Microsoft. Je peux poser des questions avec



les personnes en charge des produits qui m'intéressent.

Quels sont les avantages pour vous ?

Je peux participer à des avant-premières. Je suis invité à des événements, par exemple Imagine Cup, à Paris en juillet 2008. J'ai accès à tous les logiciels, que je reçois en version bêta.

Quels types d'actions menez-vous ?

Je suis le seul MSP de mon école, car il y a relativement peu d'informaticiens. J'ai fait une conférence sur Imagine Cup. J'en ai parlé au directeur de mon département informatique. Actuellement, je ne consacre pas beaucoup de temps pour l'organisation d'événements car il y a peu d'étudiants en informatique.

Quelques sites d'éditeurs dédiés à l'enseignement et aux étudiants :

Adobe :

<http://www.adobe.com/fr/education/>

Autodesk

www.autodesk.fr/education

www.students.autodesk.fr

Microsoft

www.microsoft.com/France/etudiants/
www.dreamspark.com

Oracle :

<http://academy.oracle.com>

www.oraclefoundation.org

www.thinkquest.org

PC Soft

www.pcsoft.fr/formation/education.htm

Autres :

<http://delegation.internet.gouv.fr/mipe>

www.syntec-informatique.fr

SALON DES FORMATIONS INFORMATIQUE & NOUVELLES TECHNOLOGIES

**DE BAC
À BAC +5**

22 MARS 2009

**PALAIS OMNISPORTS
DE PARIS BERCY
SALLE MARCEL CERDAN**

ⓂⓂ & Ⓜ¹⁴ Bercy - de 11h à 18h

Conférences
et guides
gratuits

INVITATION GRATUITE SUR WWW.STUDYRAMA.COM

Infoline 0891 36 05 28 (0,225 €/min.)

Le co-voiturage en temps réel

Ils présentaient leur projet à ImagineCup. Pour concourir avec leur logiciel de co-voiturage en temps réel, ils se sont choisis le nom de Polygeek. Ce trio de deux filles et un garçon sélectionnés pour représenter leur pays, l'Algérie, forme une équipe atypique, puisque aucun d'eux n'a fait des études d'informatique.

■ Nia Toukis



Le but du logiciel est de diminuer les embouteillages et la pollution associée en aidant les gens à se regrouper dans une voiture qui va au même endroit. Le logiciel va trouver en temps réel la voiture qui convient à ce trajet, grâce à son positionnement GPS. Le conducteur de la voiture est prévenu par SMS qu'un piéton le demande, et ce dernier peut ensuite entrer une appréciation sur son chauffeur : celui-ci est convivial ou celui-là roule trop vite.

Deux automaticiens et un futur médecin

Ils ont tenté leur chance avec enthousiasme et persévérance. ImagineCup sert aussi à cela : avoir le déclic du développement, de l'informatique ! Pour développer leur logiciel "Pick me up !", ils ont tout appris par eux-mêmes, sur le tas. "On a besoin de ça ? Comment on va le faire ? OK, on apprend le logiciel. Ça marche ? Très bien !". Amokrane Chentir, 22 ans, le développeur de l'équipe et Safa Tliba, 22 ans, étudiants en quatrième année d'automatique à l'école Polytechnique d'Alger se sont alliés avec Souad Hannane, 21 ans, en quatrième année d'études de médecine à Oran. Safa a eu son premier ordinateur à 14 ans, son amie Souad à 12 ans mais aucune d'elles ne program- mait avant la compétition mondiale Imagine Cup 2008 à Paris.

Amokrane, lui, est passionné de programmation depuis qu'il reçoit son

premier ordinateur, un Amstrad, à l'âge de 8 ans, et son premier PC à 12 ans. Il écrit son premier programme à 14 ans, en Basic. Pourquoi avoir choisi la voie de l'automatique au lieu de suivre directement des études d'informatique ? "J'ai fait l'école polytechnique d'Alger parce que c'est le meilleur diplôme qu'on peut avoir en Algérie". Mais il a bien l'intention de se rapprocher de sa passion informatique et après la coupe du monde des développeurs, il reste quelques semaines en France où il a décroché un stage dans une SSII.

Les obstacles n'ont pas manqué. Avant de se retrouver à Paris pour participer à la grande compétition mondiale, il leur a fallu passer le cap des sélections nationales en Algérie. Ils se retrouvent en lice avec dix autres équipes : toutes composées d'informaticiens, sauf la leur. Leurs collègues de l'école polytechnique d'Alger sont pessimistes. Le trio persévère et se qualifie !

Chaque étape du développement de leur logiciel les confronte avec des difficultés. Tout d'abord, il fallait trouver l'idée, cela a pris plus de trois mois. Celle-ci trouvée, le co-voiturage en temps réel, il fallait penser à l'architecture. "On voulait développer un logiciel qui permette d'agir directement, être dans l'action plutôt que dans la sensibilisation ou la mesure de la pollution." précise l'équipe. Ils répartissent les rôles : Amokrane est le développeur principal, Safa réalise les interfaces, et Souad se charge de toute la partie

design, des brochures, des animations, des vidéos. Le logiciel en temps réel est développé sur deux terminaux différents : Smartphone et PDA.

La passion vient en codant

La phase de programmation n'est pas de tout repos. D'abord, il leur faut se plonger dans les technologies et outils : Windows Vista, Windows Mobile 6.0, Visual Studio Team System 2008, SQL Server 2008 et SQL Server CE, Map Point Webservices, Windows Live id SDK, British Telecom SDK, Fonix Voice In SE 4.1 et Fonix Talk 6.0. Il fallait aussi composer avec les examens de fin d'année. Souad les a passés trois jours avant le début de la compétition mais Safa et Amokrane décident de faire l'impasse pour les passer en septembre. Mais il a fallu aussi faire les "relations publiques" : interviews, reportages, dans la presse, à la radio, à la télévision ! Du temps perdu ? Pas toujours... "Par exemple, pour le matériel, on a eu les ordinateurs personnels grâce à un sponsor- British Petroleum-, qui nous les a achetés, et c'est énorme le temps que ça prend d'écrire des rapports, d'appeler, se déplacer pour convaincre de nous aider." précise l'équipe. Aujourd'hui, Amokrane est en cours de conversion dans l'informatique, Safa s'est découvert une nouvelle passion : "au cours de ce projet, j'ai découvert le rôle de manager et j'ai vraiment apprécié. Mener un projet de A à Z, c'est un rôle que j'ai aimé tenir, je voudrais bien m'orienter dans ce domaine".

Souad veut toujours devenir médecin. ■

DÉVELOPPEZ VOTRE SAVOIR-FAIRE



Langage et code, développement web,
carrières et métier :
Programmez !, c'est votre outil
de veille technologique.

Pour votre développement personnel et professionnel,
abonnez-vous à Programmez !

Choisissez votre formule

- **Abonnement 1 an au magazine : 49 €**
(au lieu de 65,45 € tarif au numéro) *Tarif France métropolitaine*
- **Abonnement Intégral : 1 an au magazine + archives
sur Internet et PDF : 59 €** *Tarif France métropolitaine*
- **Abonnement PDF / 1 an : 30 €** - *Tarif unique*
Inscription et paiement **exclusivement en ligne**
www.programmez.com
- **Abonnement Etudiant : 1 an au magazine : 39 €**
(au lieu de 65,45 € tarif au numéro) *Offre France métropolitaine*

11 numéros par an : 49 €*

Economisez 16,45 €*

*Tarif France métropolitaine

+ Abonnement INTÉGRAL

ACCÈS ILLIMITÉ aux ARCHIVES du MAGAZINE pour 0,84€ par mois !

Cette option est réservée aux abonnés pour 1 an au magazine,
quel que soit le type d'abonnement (Standard, Numérique, Etudiant).
Le prix de leur abonnement normal est majoré de 10 € (prix identique

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

OUI, je m'abonne Vous pouvez vous abonner en ligne et trouver tous les tarifs www.programmez.com

PROGRAMMEZ

- ☐ **Abonnement 1 an au magazine : 49 €** (au lieu de 65,45 € tarif au numéro) *Tarif France métropolitaine*
- ☐ **Abonnement Intégral : 1 an au magazine + archives sur Internet et PDF : 59 €** *Tarif France métropolitaine*
- ☐ **Abonnement Etudiant : 1 an au magazine : 39 €** (joindre copie carte étudiant) *Offre France métropolitaine*

☐ M. ☐ Mme ☐ Mlle 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

Offre limitée,
valable jusqu'au
31 mars 2009

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

Conformément à la loi Informatique et
Libertés du 05/01/78, vous disposez
d'un droit d'accès et de rectification
aux données vous concernant.

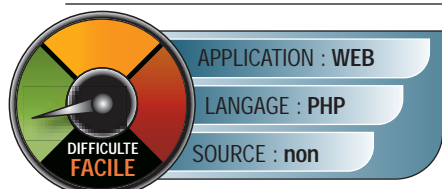
Par notre intermédiaire, vous pouvez
être amené à recevoir des propositions
d'autres sociétés ou associations.

Si vous ne le souhaitez pas, il vous
suffit de nous écrire en nous précisant
toutes vos coordonnées.

PROgrammez!

Maîtriser les fondamentaux de PHP!

PHP est un langage de scripts utilisé principalement pour produire des pages web dynamiques à l'aide d'un serveur HTTP. Disposant de plus de 3 000 fonctions natives, ce langage couvre tous les domaines en rapport avec les applications web. PHP est distribué via une licence libre qui permet sa rediffusion, son utilisation et sa modification. En d'autres termes, vous pouvez utiliser PHP librement et gratuitement.



Trois composantes sont en général utilisées, un serveur web, PHP qui est le plus souvent installé comme module du serveur web et un serveur de base de données. Il existe des

logiciels qui permettent une installation simple et rapide de cet environnement comme Wampserver sous Windows, LAMP sous Linux ou MAMP sous Mac OS X.

STRUCTURES DE BASE

L'une des forces de PHP est sa simplicité. En effet sa syntaxe est proche de langages connus comme le Java ou le C. La syntaxe de PHP hérite de Perl et de C, son modèle objet est inspiré de Java. Ce qui permet le plus souvent une prise en main (quasi) immédiate.

Balises d'ouverture

Le début et la fin des portions de code PHP d'un script sont signalés grâce à des balises d'ouverture et de fermeture. Seul ce qui est entre ces balises est interprété par PHP, le reste est envoyé tel quel au navigateur.

Les différentes balises d'ouverture et de fermeture PHP

Ouverture	Fermeture
<code><?php</code>	<code>?></code>
<code><?</code>	<code>?></code>
<code><%</code>	<code>%></code>
<code><script language="php"></code>	<code></script></code>

Nous vous conseillons fortement l'utilisation uniquement de l'ouverture `<?php`, car elle est la seule vraiment portable sur toutes les configurations. En effet, si vous utilisez la balise d'ouverture `<?`, vous risquez d'avoir un problème avec les fichiers XHTML qui commencent par la ligne `<?xml version="1.0" encoding="UTF-8"?>`, de plus, il faudra que la directive `short_open_tags()` soit activée sur le serveur. Tout ce qui dépend de la configuration du serveur est à éviter.

Insertion de PHP dans HTML

Le code PHP peut être directement intégré dans les fichiers HTML. Il peut figurer à différents endroits de ces fichiers, tout en étant entrecoupé de code HTML.

```
<html>
<head><title>Test PHP </title></head>
<body>
<p class="rouge"><?php echo 'ceci est du code PHP';?></p>
<p> ceci est aussi du <?php echo $message; ?></p>
</body>
</html>
```

Enchaînement des instructions

Les instructions PHP doivent être placées entre les balises d'ouverture et de fermeture de PHP (`<?php` et `?>`) et être séparées par des points-virgules.

```
<?php
$a = 5;
$b = 3;
$c = "PHP5";
echo $c;
?>
```

Seuls ces points-virgules séparent les différentes instructions ; les retours à la ligne n'ont aucune influence. Il est donc possible d'imbriquer plusieurs instructions sur la même ligne ou de faire une unique instruction sur plusieurs lignes :

```
<?php
$a = 5;
$b = 3;
$c = "PHP5";
echo $c;

/* Équivalent à */

$a = 5; $b = 3;
$c
=
    "PHP5"; echo $c;
?>
```

Cette dernière notation, bien que possible, est fortement déconseillée car elle entraîne d'importantes difficultés de relecture et favorise l'apparition d'erreurs.

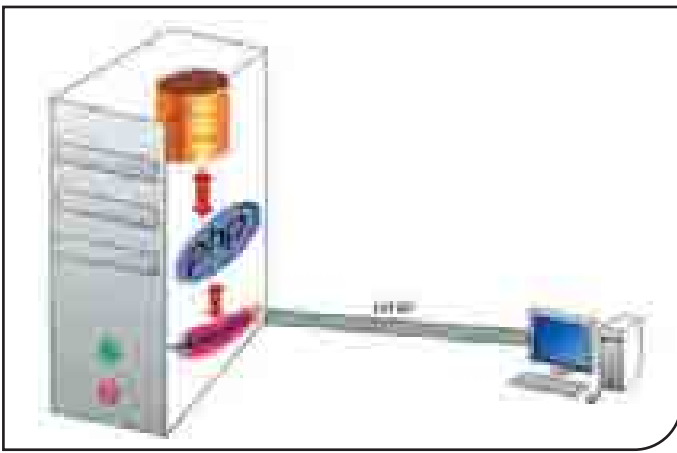
Affichage et envoi de code HTML

La commande **echo** indique au serveur qu'il doit renvoyer les informations contenues entre les guillemets. On notera qu'il est possible de renvoyer du code HTML (balises `<p>` et `</p>`). On peut tout aussi bien renvoyer le code HTML désignant une image :

```
<?php
echo "<p>Ceci est du code PHP</p> ";
echo "<img src='../image.gif' alt=''>";
?>
```

Exécuter du code PHP

Dans une configuration web client/serveur, pour exécuter votre code PHP, il faut appeler votre script en tapant l'adresse du serveur suivi du nom du script via un navigateur web. Les scripts appelés



devront tous avoir une extension reconnue par le serveur web comme appartenant à PHP généralement **.php**. Ce comportement peut être modifié en touchant à la configuration du serveur web (ex : Apache), avec de la réécriture d'URL notamment.

Pour un script local : http://localhost/mon_script.php

Pour un script distant : http://adresse_web/mon_script.php

LES VARIABLES

Une variable peut être représentée comme un récipient disponible pendant toute l'exécution de votre programme. Ainsi, au cours du script, vous pouvez lui donner des valeurs, les modifier et les utiliser. En PHP, l'utilisation des variables est très simple et ne nécessite aucune déclaration préalable. Les variables en PHP se trouvent sous la forme `$nom_variable`. Elles commencent par le symbole `$` et sont formées d'une suite de lettres, de chiffres et de caractères de soulignements. Elles sont sensibles à la casse et le premier caractère du nom d'une variable ne peut pas être un chiffre.

```
<?php
$nom_client = "freshdaz";
$age        = 34;
$tel1       = "01.45.28.09.82"
?>
```

Les noms de variable

Correct	Incorrect	Explication
<code>\$Variable</code>	<code>\$Variable 1</code>	Contient des espaces
<code>\$variable</code>	<code>Variable</code>	Une variable commence toujours par \$
<code>\$variable_double</code>	<code>\$variable-double</code>	Le signe - est interdit
<code>\$variable_email</code>	<code>\$test@yahoo.fr</code>	Les caractères @ et . sont interdits.
<code>\$test2</code>	<code>\$2test</code>	Une variable ne commence pas par un chiffre.

LES CONSTANTES

Les constantes sont un autre type de récipient. Elles sont souvent utilisées pour gérer des paramètres de configuration ou des traductions de manière centralisée. On les définit à l'aide de la fonction `define()`. Elles ne peuvent plus par la suite recevoir d'autres valeurs. Par convention, pour faciliter la relecture du code, on écrit les constantes en majuscules. Contrairement aux variables, les constantes ne comportent pas de `$` devant leur nom.

```
<?php
define("ORGANISME_DE_FORMATION", "Anaska Alter Way");
echo ORGANISME_DE_FORMATION;
?>
```

LES TYPES DE DONNÉES

PHP dispose de quatre types de données simples : des booléens, des entiers, des nombres décimaux et des chaînes de caractères. PHP donne dynamiquement un type à la variable selon la valeur que vous lui assignez. Ce type peut changer au cours de l'exécution selon les valeurs affectées.

Booléens (boolean)

Un booléen est une valeur pouvant être soit vraie, soit fausse. Le mot-clé **TRUE** désigne un booléen vrai, et le mot-clé **FALSE** un booléen faux. Ces mots-clés sont insensibles à la casse.

```
<?php
$mon_bool_vrai = true; // Booléen vrai
$mon_bool_faux = FALSE; // Booléen faux
?>
```

Les nombres entiers (integer)

Les nombres entiers peuvent être entrés tels quels dans le code. Les entiers négatifs sont à précéder du symbole `-`.

Un entier commençant par un chiffre de 1 à 9 sera interprété selon la base décimale habituelle. C'est la notation que vous utilisez tous les jours. S'il commence par un zéro, il sera compris en base octale (015 sera interprété comme le nombre 13 en base décimale). S'il commence par 0x, il sera interprété selon la base hexadécimale (0x1A sera interprété comme le nombre 26 en base décimale).

```
<?php
$nombre = 45;
$nb negatif = -15;
$nb_hexa = 0x1A;
?>
```

Les nombres décimaux (double, float)

Les nombres décimaux sont aussi interprétés directement par le moteur PHP. Ce sont des nombres plus grands que ne peut l'être un entier (un peu plus de 4 milliards sur les systèmes 32 bits), ou comportant une partie décimale. Un nombre décimal comporte soit un point (équivalent anglais de la virgule), soit un `" e "` (majuscule ou minuscule) séparant l'exposant.

```
<?php
$nombre = 3.14159;
$nombre = 5e7;
$nombre = 1.000;
?>
```

Les nombres à virgule flottante sont par définition imprécis. Pour les gérer, PHP fait des approximations. Vous ne devriez normalement pas vous en soucier car l'erreur d'approximation est négligeable, mais il peut être important de le remarquer si vous comparez deux nombres à virgule flottante : même si pour vous ils sont égaux, pour PHP, ce n'est pas forcément le cas à cause des erreurs d'arrondis et d'approximation.

Les chaînes de caractères (string)

Les chaînes de caractères sont généralement délimitées par des guillemets.

```
<?php
$chaine = "PHP 5 avancée est un livre référence";
echo $chaine;
?>
```

La taille d'une chaîne de caractères (le nombre de signes, espaces et caractères blancs compris) est donnée par la fonction `strlen()`.

```
<?php
$livre = 'PHP 5 avancé';
```

```
echo strlen($livre); // Affiche 12
?>
```

Si vous utilisez Unicode, certains caractères peuvent utiliser plus d'un octet sur le disque ou en mémoire. Dans ce cas, c'est la taille en octets et non en caractères qui est retournée par `strlen()`. La chaîne " Éric " retournera donc une taille de 5 et non de 4 si vous utilisez un codage UTF-8 (le E accentué prend deux octets).

Il vous est aussi possible de trouver le nombre de mots d'une chaîne grâce à la fonction `str_word_count()`.

```
<?php
    $livre = 'PHP 5 avancé';
    echo str_word_count($livre); // Affiche 3
?>
```

LES TABLEAUX (ARRAY)

En plus des types de données simples, PHP propose une façon de grouper ces données : les tableaux. PHP permet deux types de tableaux : les tableaux indexés numériquement et les tableaux associatifs. Il n'est pas nécessaire de déclarer leur taille lors de la déclaration, elle est gérée dynamiquement par PHP.

Tableaux indexés numériquement

Un tableau indexé est une simple liste d'éléments. On peut le créer grâce au mot-clé `array()`, en séparant les valeurs par des virgules.

```
<?php
    $tableau = array(12250, 15555, 12000, 21300, 25252, 20010, 8460);
    $stab2    = array($variable, "texte", 153);
?>
```

Dans la liste, chaque élément est repéré par sa position, son index. Cet index sera bien entendu unique. Chacune de ces variables est rangée dans une case. Pour y accéder, il faut indiquer le nom du tableau et la case (l'index) de la variable désirée.

La syntaxe est la suivante : `$nom_variable_tableau[index]`

On notera que les index commencent à partir de 0 et non de 1. Le premier élément d'un tableau est `$tableau[0]`.

Il est possible de lire et écrire directement dans une case grâce à cette syntaxe :

```
<?php
    // On commence par assigner des valeurs au tableau
    $tableau[0]    = 12250;
    $tableau[10]   = 35120;

    // Puis on peut le manipuler
    echo $tableau[10];
?>
```

Il existe une syntaxe réduite pour ajouter un à un les éléments sans avoir à manipuler les index. PHP numérottera automatiquement la case, en incrémentant l'index numérique. Il suffit d'omettre l'index, mais en laissant les crochets. Ces trois codes sont équivalents :

```
<?php
    $stabA = array(1, 2, 3);

    $stabB[0] = 1; $stabB[1] = 2; $stabB[2] = 3;
```

```
$stabC[] = 1; $stabC[] = 2; $stabC[] = 3;
?>
```

Tableaux associatifs

Les tableaux numériques indexés sont faciles d'utilisation, mais peuvent se révéler peu pratiques pour gérer la signification et la place des valeurs contenues. Effectivement, nous devons forcément passer par le numéro correspondant à un élément. L'alternative proposée par PHP est le tableau associatif. Celui-ci associe une chaîne de caractères à un élément. On parle alors de tableau associatif.

Pour créer un tableau grâce au mot-clé `array()`, il faut donner la clé et l'élément, séparés par `=>`.

```
<?php
    $stab = array(
        "prenom" => "Cyril",
        "ville"  => "Paris",
        "travail" => "informatique"
    );
?>
```

Comme pour les tableaux indexés, il est possible de référencer directement un élément grâce à sa clé. On peut alors considérer que les tableaux indexés ne sont que des tableaux associatifs dont les clés sont numériques et attribuées automatiquement.

```
<?php
    $tableau['nom'] = "PIERRE de GEYER";
    $tableau['prenom'] = "Cyril";
    $tableau['sexe'] = "m";
    $tableau['ville'] = "Paris";
    $tableau['cp'] = "75005";
    $tableau['telephone'] = "0143819291";
    $tableau['travail'] = "informatique";
    $tableau['Pays'] = "france";

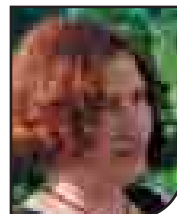
    echo "Votre contact à {$tableau['ville']} sera monsieur ";
    echo $tableau['nom'];
    echo "<br>Son téléphone est le ";
    echo $tableau['telephone'];
?>
```

PHP dispose d'un nombre important de fonctions natives pour manipuler les tableaux, et nativement gère certaines données sous cette forme (formulaires, sessions) c'est pourquoi maîtriser cette structure est indispensable. Merci à **Cyril PIERRE de GEYER** pour ses précieux conseils.



■ Olivier DASINI

Consultant et formateur officiel certifié MySQL. Il est responsable du pôle SGBDR chez Anaska Alter Way Group. Auteur de nombreux articles sur MySQL, il est également co-fondateur du MySQL User Group Francophone : LeMUG.fr.



■ Sarah Haïm-Lubczanski

Consultante et formatrice certifiée PHP, Ajax, Javascript et OpenOffice chez Anaska Alter Way Group. Elle est également vice-secrétaire de l'AFUP (Association Française des Utilisateurs de PHP) et trésorière de l'association Ergolibre.

Les outils de la Direction Informatique

*Vous avez besoin d'info
sur des sujets d'administration,
de sécurité, de progiciel,
de projets ?
Accédez directement
à l'information ciblée.*



L'INFORMATION SUR MESURE

Actu triée par secteur Cas clients Avis d'Experts

Etudes & Statistiques

Infos des SSII

Vidéos

Actus Evénements Newsletter

The image is a screenshot of the website 'www.solutions-logiciels.com'. It features a navigation bar at the top with links for 'Actu triée par secteur', 'Cas clients', and 'Avis d'Experts'. The main content area is divided into several sections, including 'Etudes & Statistiques', 'Infos des SSII', 'Vidéos', 'Actus', 'Evénements', and 'Newsletter'. Red lines are drawn from the labels on the right to the corresponding sections on the website screenshot.

L'INFORMATION EN CONTINU

www.solutions-logiciels.com





L'architecture orientée développeur

Débuter avec BizTalk Server 2009

Dans la première partie de cet article, nous avons vu comment développer une petite application BizTalk. Ce mois-ci, nous allons continuer à découvrir des fonctionnalités de BizTalk Server, et plus particulièrement l'exposition de web services et l'annuaire UDDI. Nous repartirons donc de l'application précédente afin de l'exposer en web service. La démarche restera inchangée, à savoir : vous fournir une description en pas à pas de manière à vous permettre de rejouer l'ensemble sur votre environnement BizTalk.



Tout d'abord, nous devons nous assurer que l'application développée dans la première partie est installée et opérationnelle.

Pour cela, le plus simple est de réinstaller la solution à partir du fichier disponible en téléchargement : "stock-Partie2-Init.zip". Une fois le fichier décompressé, vous disposez :

- Du code source de la partie 1
- De scripts d'installation

Exécutez alors le fichier de commande "InstallAll.cmd" : celui-ci se charge d'installer l'application et de créer la base de données.

Il ne reste alors plus qu'à démarrer l'application dans la console

d'administration BizTalk (Voir article précédent) puis à tester (à cet effet, deux fichiers exemples sont fournis).

Il suffit de déposer une copie dans le répertoire `c:\in` pour déclencher le traitement).

EXPOSITION EN SERVICE WEB WCF

Nous allons maintenant, à l'aide de l'assistant de publication WCF, créer un service web qui servira de point d'entrée pour notre processus BizTalk. Cet assistant va se charger de générer et publier un service selon nos spécifications. A l'issue de cette étape, il nous res-

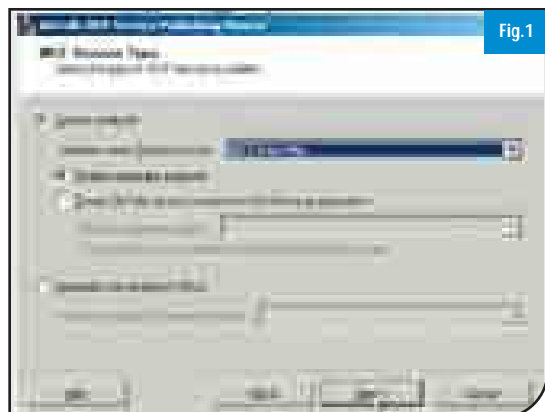


Fig.1



Fig.2

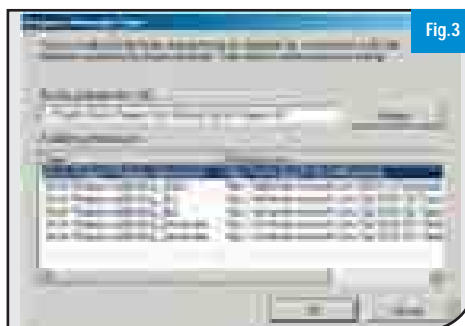


Fig.3

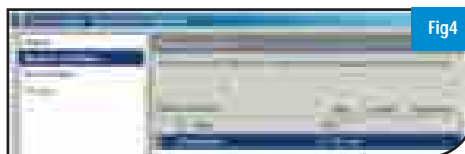


Fig.4

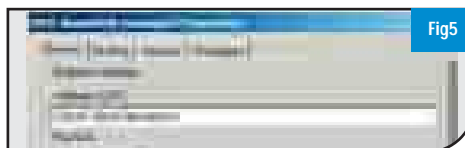


Fig.5

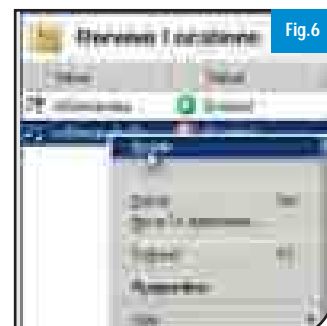


Fig.6



Fig.8

tera à autoriser ce mode de communication en créant une nouvelle " Receive Location " sur le port d'entrée utilisé par notre orchestration. Nous n'aurons donc aucun " code " à écrire.

Le lancement de l'assistant de publication (BizTalk WCF Service Publishing Wizard) peut se faire soit depuis Visual Studio (Menu Tools), soit directement depuis le dossier BizTalk Server 2009, dans le menu démarrer. Une fois lancé, et passé l'écran d'accueil, nous arrivons sur un écran dans lequel nous devons définir les paramètres du service et notamment la pile de protocole à utiliser (Binding) : [Fig.1]

Ici il faut spécifier les paramètres suivants :

- Adapter name : WCF-BasicHttp
- Enable metadata endpoint : coché
- Create BizTalk receive locations : non coché

Une fois ceci fait, nous pouvons continuer l'assistant. A l'étape suivante nous devons choisir entre deux options : exposer une orchestration ou exposer des schémas. L'exposition d'une orchestration est une option rapide permettant de créer le web service à partir d'un port d'orchestration, mais dans notre cas cette option n'est pas la mieux adaptée car elle ne permet pas, simplement, d'autoriser la réception selon plusieurs protocoles (Pour nous : File et WCF). Nous choisirons donc la deuxième option.

La suite de l'assistant nous permet alors de configurer complètement le nom du service, ses opérations et le type de message pour chaque opération : ceci se fait au travers de menus contextuels res-

pectivement au niveau Service/Opération/Requête et Réponse. Dans cet assistant nous allons effectuer les opérations suivantes :

- Renommer le service (Service1) en StockService
- Supprimer l'opération initiale (Operation1) et la remplacer par une opération unidirectionnelle (Menu contextuel : Add Web Method / One Way) que nous renommerons ensuite " DemandeReapro "
- Au niveau du paramètre Request de la méthode DemandeReapro, nous sélectionnons dans le menu contextuel " Select Schema Type " : [Fig.2]

Nous devons alors pointer vers l'assembly de notre projet (Stock.Reapro.dll) et sélectionner le bon type de message (<http://www.btug.fr/Stock#Demande>) : [Fig.3]

Le contrat de notre service est maintenant défini. L'étape suivante de l'assistant permet de définir le namespace associé au web service (ex : <http://www.masociete.com/Stock>). Ensuite nous rentrons dans les options de déploiement IIS (Internet Information Services, le serveur web de Windows) :

- Spécifiez l'emplacement du web service : <http://localhost/stock>
- Autorisez l'accès anonyme (case à cocher Allow anonymous access ...)

Nous pouvons maintenant compléter les dernières étapes de l'assistant. Un service a été créé sous IIS mais n'est pas pour l'instant opérationnel : il reste une étape de configuration à effectuer dans BizTalk. →



CODit SGA Dashboard
CODit Collaboration Integration Platform
CODit BizTalk Implementation Framework



Business Process and Integration
MS/Software Solutions

Un centre de compétences BizTalk

Montez en compétences au sein d'une équipe de renommée internationale en plein cœur de Paris



CONFIGURATION DU SERVICE DANS BIZTALK

Notre application (Stock) dispose d'un port de réception (recvDemandes) qui permet à l'orchestration de recevoir les demandes de réapprovisionnement. Nous allons lui rajouter une "Receive Location" qui correspondra au service WCF que nous venons de créer.

Pour effectuer cette opération :

- Lancez la console d'administration, déployez le groupe Biztalk et l'application Stock
- Dans les ports de réception, double-cliquez sur le port recvDemandes afin d'afficher ses propriétés
- Sélectionnez "receive location" puis "new" : [Fig.4]

Dans les paramètres généraux de la receive location rentrez :

- Le nom : rcldemandesFromWS
- Le type : WCF-BasicHttp
- Le pipeline de réception : XMLReceive

Sélectionnez maintenant configure afin de rentrer les paramètres pour l'adaptateur WCF-BasicHttp.

Dans notre cas, le seul paramètre à renseigner est l'adresse du service. Notez qu'ici il faut rentrer une adresse relative : /Stock/StockService.svc [Fig.5]

Nous avons fini la configuration. Il reste à valider et activer la receive location (Menu contextuel / Enable) : [Fig.6]

CONFIGURATION IIS

Il nous reste une dernière étape de configuration avant que le service soit pleinement opérationnel. En effet notre service a été publié dans IIS avec la configuration par défaut, or, cette configuration ne permet pas au service de dialoguer avec BizTalk. Pour changer ceci, nous devons créer un nouveau "Application Pool" dans IIS. Cet "Application Pool" représente le processus dans lequel s'exécute notre service et est associé au compte Windows : ce compte doit avoir les droits suffisants pour accéder à BizTalk.

Pour effectuer cette configuration, il faut tout d'abord lancer la console d'administration de IIS (Depuis le menu Start → Administrative Tools → Internet Information Services (IIS) Manager).

Dans la console d'administration, après avoir navigué jusqu'au dossier "Application Pools", nous pouvons en créer un nouveau en choisissant "new" dans le menu contextuel : [Fig.7]

Après avoir donné un nom (BizTalkWSAppPool) à notre application pool, le plus simple est de copier les paramètres d'un autre qui est correctement configuré : BAMAppPool.

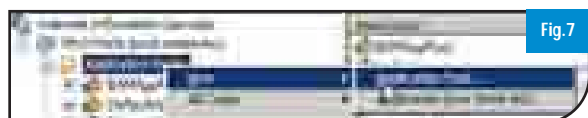


Fig.7



Fig.11



Fig.12



Fig.9



Fig.10



Fig.13

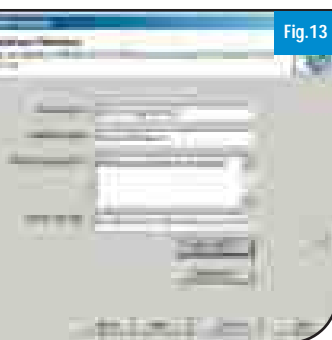


Fig.13

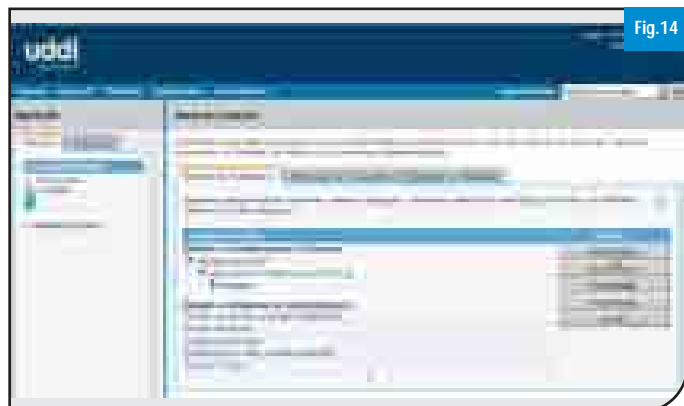


Fig.14

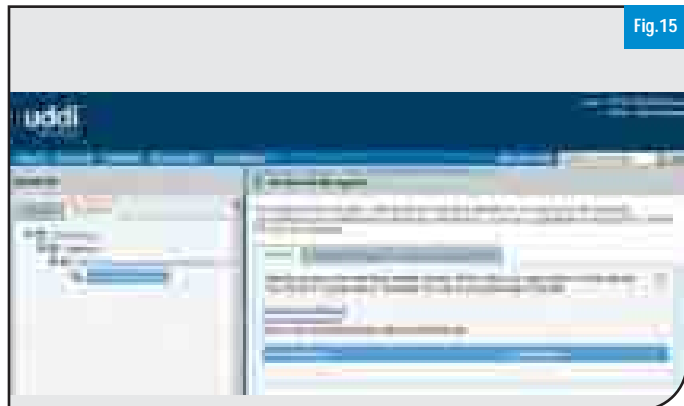


Fig.15

Nous allons maintenant modifier les propriétés du répertoire virtuel correspondant à notre application (Il se trouve dans la console IIS dans Web Site / Default Web Site / Stock : accès aux propriétés par le menu contextuel : Properties) : [Fig.8]

Il faut alors sélectionner notre nouveau "Application Pool" sur le premier onglet de l'écran de propriété.

Si tout s'est bien passé, on peut naviguer avec Internet Explorer vers l'url <http://localhost/stock/StockService.svc>, ce qui devrait afficher une page de description du service. [Fig.9]

Pour tester le service, vous pouvez, par exemple créer un petit projet Windows Form tel que celui-ci : [Fig.10]

(NB : le source complet de la solution ainsi que le client de test sont disponibles en téléchargement)

PUBLICATION DANS L'ANNUAIRE UDDI

BizTalk 2009 est livré avec un annuaire de service UDDI qui permet de recenser et de classifier les web services disponibles au sein de l'entreprise. Nous allons voir comment publier notre nouveau service dans cet annuaire.

La manière la plus simple d'effectuer cette publication est d'utiliser l'assistant UDDI. On le trouve dans le menu Start → Microsoft UDDI Services → UDDI Tools → Publishing Wizard.

Une fois l'assistant lancé, les premières étapes sont les suivantes :

- Préciser le type d'authentification utilisé pour accéder à l'annuaire : Windows Integrated Authentication
- Fournir l'url du site de publication : <http://localhost/uddi/publish.aspx>

Nous arrivons alors sur un écran qui va nous permettre de créer un " Provider " : il s'agit en fait de l'éditeur ou du département au sein de l'entreprise qui fournit le service : [Fig.11]

Nous pouvons ensuite classer le provider (bouton Categorize). L'annuaire est livré avec un ensemble de catégories prédéfinies, mais il nous est possible de définir les nôtres (NB : cette fonction est accessible à l'aide d'un outil spécifique : le " Categorization Scheme Editor " dans UDDI Tools) : [Fig.12]

Une fois le provider créé, nous pouvons enregistrer les informations concernant notre service ainsi que l'emplacement du fichier de description (WSDL). [Fig.13]

Il est également possible, et c'est fortement conseillé, de classer notre service.

Nous pouvons alors terminer l'assistant et les entrées auront été créées dans l'annuaire UDDI.

Pour consulter l'annuaire et confirmer la présence de notre service nous pouvons utiliser le site web UDDI qui se trouve à l'adresse : <http://localhost/uddi>

Nous avons alors la possibilité, par exemple de rechercher par catégories : [Fig.14 et 15]

Ceci termine cette deuxième partie sur BizTalk Server 2009. N'hésitez pas à expérimenter avec les services WCF BizTalk et l'annuaire UDDI ...

■ Roch Baduel - MVP Biztalk

Responsable pôle intégration / SOA – MCNEXT. rbaduel@mcnext.com



En mars, cet informaticien choisira son futur employeur !

Salon Lesjeudis.com
INFORMATIQUE - RECRUTEMENT

5 mars - NANTES
Comptabilité - Développement - Support

26 mars - GENEVE
Comptabilité - Développement - Support

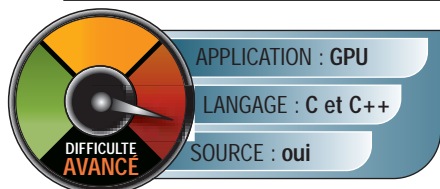
7 avril - AUVERGNE-PROVENCE
9 avril - PARIS
14 mai - BORDEAUX
11 juin - TOULOUSE

24 septembre - PARIS
Octobre - NANTES
26 Novembre - LYON

Renseignements et pré-inscription sur www.lesjeudis.com

Découvrir et mettre en œuvre CUDA 1^{re} partie

Avec la librairie CUDA, votre accélérateur graphique NVidia n'est plus un simple accélérateur graphique, il devient un accélérateur de calcul généraliste. Et quel accélérateur... Découverte et notions de base.



L'air du temps est aux processeurs multi-cœurs et à la programmation parallèle. Une répartition de la charge de travail sur les cœurs des nouveaux processeurs

est donc incontournable. Incontournable ? Pas forcément. Tous les PC modernes sont équipés d'une puce extraordinairement puissante et finalement assez méconnue : la GPU. Un composant qui doit en avoir sous le pied pour être capable d'animer les jeux 3D dernier cri. Et si au lieu de se limiter à se détendre avec le dernier Shoot'em up, on utilisait cette puissance pour décharger l'unité centrale, la CPU, en demandant à la GPU d'effectuer des calculs lourds, en parallèle qui plus est ? Telle est l'idée qui préside à CUDA, une librairie de programmation en langage C écrite par Nvidia pour exploiter les fonctionnalités de calcul des GeForce 8 (ou supérieures) directement à travers leur driver. Par cet article, nous allons nous initier à la programmation CUDA, un univers réellement fascinant dans lequel vous découvrirez que votre PC peut très facilement se transformer en monstre de calcul. Et le terme n'est pas exagéré. Ainsi en va-t-il pour le projet Folding@Home, un projet de calcul réparti qui a pour but d'étudier le repliement de protéine dans diverses configurations de température et de pression afin de mieux comprendre ce processus et d'en tirer des connaissances utiles relatives entre autres à la maladie d'Alzheimer et à certains types de cancers. Depuis que le moteur de calcul a été porté sur CUDA les temps de calculs ont été réduits par 100 ! Les possesseurs d'une carte graphique ATI disposent quant à eux de la librairie CTM qui ne fait pas l'objet de cet article.

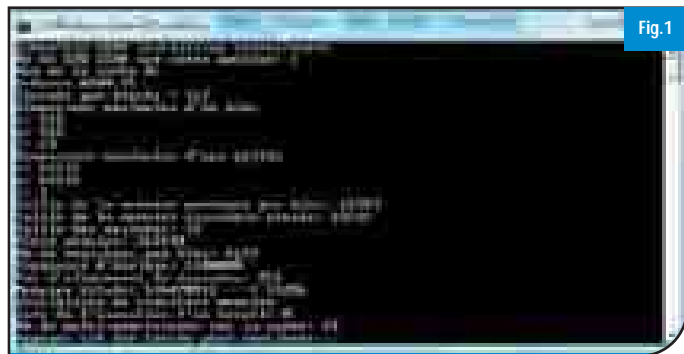
1 LES OUTILS NÉCESSAIRES

En plus du magazine Programmez ! qui est toujours nécessaire :) vous devez avoir un PC ou un Mac (Mac OS X 10.5.2+) équipé d'une carte graphique comportant une puce Nvidia GeForce 8 ou supérieure, ou si vous avez les moyens, d'une puce Tesla ou Quadro. Quel que soit le modèle, celui-ci doit être accompagné de 256 Mo de Ram au minimum. Moyennant quoi tout code CUDA doit fonctionner sous toutes plates-formes. Les exemples de cet article ont été essayés avec succès sous Windows Vista, avec une GeForce 8800 GT et une 8400M GS (ordinateur portable). Sous Windows vous devez avoir Visual Studio 2005. Cela ne fonctionne pas avec Visual Studio 2008. Sous Windows, il est encore possible d'utiliser le compilateur MinGW ou encore de compiler sous l'environnement Cygwin. Vous devez avoir installé un driver très récent. Le mieux est encore d'installer celui proposé à la page de téléchargement CUDA pour votre plate-forme. Vous devez installer le toolkit CUDA qui contient le compilateur nvcc et les librairies CUDA. Le compilateur est un frontal à votre compilateur C ou C++ habituel. Son rôle est de transformer le code CUDA en un code C qui sera ensuite compilé de

manière classique. Enfin on ne se privera en aucun cas d'installer la SDK CUDA qui contient moult exemples et documentations à étudier absolument. Toutes ces installations ne présentent pas de difficulté et nous ne nous y intéresserons pas dans cet article. Enfin pour travailler avec CUDA vous devez être à l'aise avec C ou C++, notamment en ce qui concerne les pointeurs.

2 MULTI-PROCESSEURS, HIÉRARCHIES DE THREADS ET PROGRAMMATION PARALLÈLE

Se lancer dans CUDA c'est se lancer dans la programmation parallèle. En jargon CUDA, une routine devant s'exécuter sur la GPU s'appelle un kernel. Quand un kernel est invoqué, l'exécution de celui-ci est automatiquement threadée. Jusque là rien de très étonnant. Mais la spécificité de CUDA est d'exécuter simultanément des centaines de threads gérés au niveau du matériel et c'est ce qui explique pourquoi les performances peuvent être autant accélérées, pourvu que le programmeur sache profiter de tout cela en décomposant judicieusement un calcul en une multitude de sous-tâches. Comment cela se passe-t-il ? Au niveau hardware, une GPU est constituée d'un certain nombre de multi-processeurs. Un multi-processeur est composé de huit processeurs disposant chacun de registres et partageant de la mémoire cache ou read-only (lecture seule). Un multi-processeur traite les threads par fournée de 32, ce qui s'appelle une voilure (warp) dans le jargon CUDA. L'exécution des 32 threads est dispatchée sur les 8 processeurs internes. D'après la documentation, comme toute cette gestion, y compris la synchronisation des threads, est effectuée au niveau matériel, le coût de la planification est nul. Une GeForce 8800 GT comporte 14 multi-processeurs, ce qui fait 112 processeurs. Nous sommes fort loin des 2 ou 4 cœurs d'une CPU et l'on commence à entrevoir pourquoi CUDA apporte tant de puissance. Voilà pour le côté matériel. Côté logiciel, donc côté programmation, c'est tout aussi ébouriffant. La philosophie n'est pas d'invoquer les kernels, et donc de lancer les threads un par un mais par blocs. Mieux que cela, les blocs, qui peuvent être compris comme des tableaux de threads peuvent être à une, deux ou trois dimensions. Ce n'est pas tout, les blocs de threads peuvent



Les caractéristiques de notre Nvidia 800 GT selon CUDA.

être à leur tour groupés dans des grilles à deux dimensions.... C'est le runtime qui va se charger d'éclater l'ensemble des threads pour les répartir sur les multi-processeurs. Le programmeur n'a jamais à s'occuper de cette question. (Heureusement... ;)

3 HELLO WORLD CUDA

Avant de nous lancer dans les calculs impliquant des centaines de threads, familiarisons-nous avec le maniement de CUDA et apprenons à compiler une application. Pour cela, écrivons un Hello World qui affichera les caractéristiques de la GPU sur la console. [Fig.1]

```
include <cuda.h>

#include <iostream>
using namespace std;

int main(int argc, char** argv)
{
    // Initialiser CUDA
    CUresult result = cuInit(0);
    switch(result)
    {
        case CUDA_SUCCESS:
            cout << "Librairie CUDA initialisee correctement" << endl;
            break;
        case CUDA_ERROR_INVALID_VALUE:
        case CUDA_ERROR_NO_DEVICE:
        default:
            cout <<
            "Une erreur s'est produite à l'initialisation de CUDA" << endl;
            return EXIT_FAILURE;
    }

    // Combien de cartes graphiques Nvidia/CUDA sur la machine ?
    int nb_gpu;
    cuDeviceGetCount(&nb_gpu);
    cout << "Nb de GPU CUDA sur cette machine: " << nb_gpu << endl;

    // Obtenir un Handle sur la carte 0
    CUdevice device;
    cuDeviceGet(&device, 0);

    // Obtenir le nom de la carte 0
    char buffer[256];
    cuDeviceGetName(buffer, sizeof(buffer), device);
    cout << "Nom de la carte 0:" << endl;
    cout << buffer << endl;

    // Obtenir les propriétés de la carte 0
    CUdevprop prop;
    cuDeviceGetProperties(&prop, device);
    cout << "Threads par blocks : " << prop.maxThreadsPerBlock << endl;

    cout << "Dimensions maximales d'un bloc" << endl;
    cout << "x: " << prop.maxThreadsDim[0] << endl;
    cout << "y: " << prop.maxThreadsDim[1] << endl;
    cout << "z: " << prop.maxThreadsDim[2] << endl;
```

```
// etc... Voir site

cout << "Taille de la memoire partagee par bloc: "
    << prop.sharedMemPerBlock << endl;
// etc... Voir site
// Enfin obtenir deux propriétés qui ne
// figurent pas dans CUdevprop
int value;
cuDeviceGetAttribute(&value,
    CU_DEVICE_ATTRIBUTE_GPU_OVERLAP, device);
cout << "Possibilite de transfert memoire" << endl;
cout << "lors de l'execution d'un kernel: " << value << endl;

cuDeviceGetAttribute(&value,
    CU_DEVICE_ATTRIBUTE_MULTIPROCESSOR_COUNT, device);
cout << "Nb de multi-processeurs sur la carte: " << value << endl;

return 0;
}
```

Ce code C++ démontre l'utilisation des APIs CUDA pour obtenir tous les renseignements relatifs à votre ou vos GPUs. Le code étant répétitif et ne présentant aucune difficulté, nous n'en avons donné qu'un extrait (Vous trouverez le code complet sur notre site www.programmeur.com). Nous nous demandons bien pourquoi nous devons utiliser deux APIs, `cuDeviceGetProperties` et `cuDeviceGetAttribute` alors qu'une seule aurait très bien pu faire l'affaire, mais c'est ainsi. L'édition de liens doit être faite avec le fichier `cuda.lib` ou `cuda.so` selon votre plate-forme. Enfin on remarque, au début du code, l'appel à `cuInit` qui doit figurer au début de toute application CUDA, sans exception.

4 PREMIER CALCUL CUDA

Notre objectif est très modeste, nous allons demander à notre GPU d'incrémenter les coordonnées d'un vecteur que nous lui fournirons. Cet objectif est tellement modeste que l'opération serait plus rapide sans passer par CUDA, mais nous sommes en phase d'apprentissage... Nous employons cette fois le langage C. Créez sous Visual Studio un projet de type console, de nom par exemple `IncVector`, et renommez le fichier `IncVector.cpp` en `IncVector.cu`. Si vous travaillez sans IDE, par exemple sous Linux, créez le fichier directement. Le lecteur n'aura aucun mal à transposer sous Linux les instructions que nous donnons maintenant sous Visual Studio uniquement. Le fichier `.cu` est destiné à recevoir un mélange de code C et de code CUDA spécifique. Il devra donc être compilé par le compilateur `Nvidia nvcc`. L'édition de liens se fera de manière classique. Pour pouvoir utiliser `nvcc` sous Visual Studio, ouvrez le dialogue des propriétés du fichier `IncVector.cu` et dans "Etape de génération personnalisée|Généralé, Renseignez "Ligne de commande" ainsi:

```
"$(CUDA_BIN_PATH)\nvcc.exe" -ccbin "$(VCInstallDir)bin" -c -D
_DEBUG -DWIN32 -D_CONSOLE -D_MBCS -Xcompiler /EHsc,/W3,/nologo
,/Wp64,/Od,/Zi,/RTC1,/MTd -I"$(CUDA_INC_PATH)" -I./ -o $(Confi
gurationName)\incvector.obj IncVector.cu
```

N'oubliez pas ensuite d'ajouter:

```
$(ConfigurationName)\incvector.obj
```

deux lignes plus bas dans le dialogue, comme illustré ci-contre. [Fig.2]
 Déboguer du code CUDA est évidemment impossible puisque tout se passe dans la GPU. Mais en cas de grosses difficultés, si vous ne pouvez vous passer d'une séance de débogage, ajoutez

```
-deviceemu
```

à la commande de compilation. Votre programme s'exécutera alors dans un émulateur. Adieu performances, adieu grand nombre de threads, mais vous y verrez plus clair car vous pourrez alors utiliser le débogueur.

Dans tous les cas on donnera `cuda.lib` et `cudart.lib` en entrée à l'éditeur de liens. [Fig.3]

5 ORGANISATION D'UNE APPLICATION CUDA

Le schéma est toujours le même. On réserve de la mémoire dans la GPU, on copie des données dans cette mémoire réservée, on lance les threads, on récupère les résultats du calcul et on libère la mémoire.

Cette démarche peut être répétée autant de fois que l'on veut dans une même application dont le flux sera normal, séquentiel, sur la CPU, en dehors des phases CUDA parallélisées. Voici le code qui incrémente un vecteur :)

```
// projet IncVector sur le site
#include <stdio.h>

#include <cuda.h>
#include <vector_types.h>

void init()
{
    // Initialiser CUDA
    CUresult result = cuInit(0);
    switch(result)
    {
        case CUDA_SUCCESS:
            puts("Librairie CUDA initialisee correctement");
            break;
        case CUDA_ERROR_INVALID_VALUE:
        case CUDA_ERROR_NO_DEVICE:
```

```
default:
    puts("Une erreur s'est produite à l'initialisation de CUDA");
    exit(1);
    }
    cudaSetDevice(0);
}

__global__ void incVec(int3 *v)
{
    v->x = threadIdx.x;
    v->y = threadIdx.x;
    v->z = threadIdx.x;
}

void run(int3 *v)
{
    int3 *gpu_v;
    int size;

    size = sizeof(int3); // et non sizeof(gpu_v) Attention :)
    cudaMalloc((void**)&gpu_v, size);
    cudaMemcpy(gpu_v, v, size, cudaMemcpyHostToDevice);

    incVec<<<1, 1>>>(gpu_v);

    cudaMemcpy(v, gpu_v, size, cudaMemcpyDeviceToHost);
    cudaFree(gpu_v);
}

void print_vector(int3 *v)
{
    printf("x: %d, y: %d, z: %d\n", v->x, v->y, v->z);
}

int main(int argc, char** argv)
{
    int3 v;

    init();
    v = make_int3(-1, -1, -1);
```

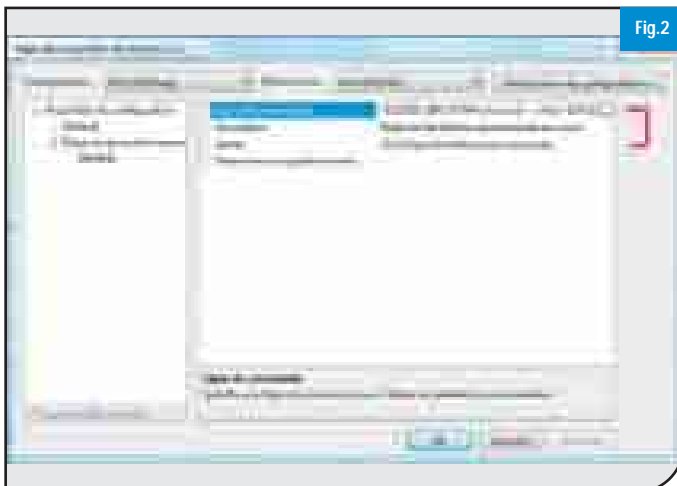


Fig.2

Configuration de la compilation avec `nvcc` sous Visual Studio.

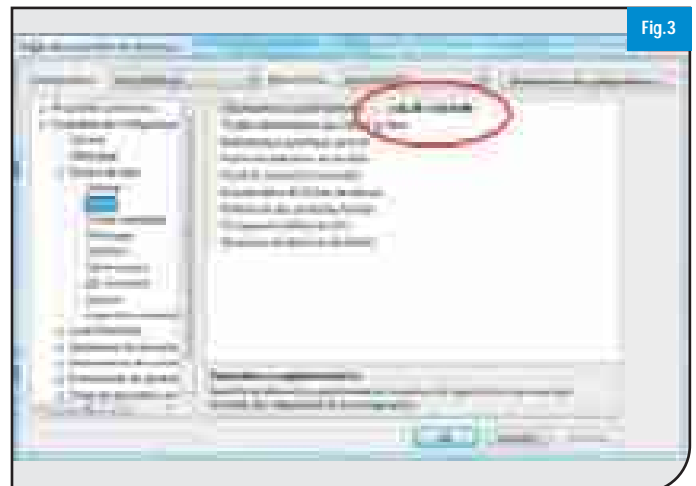


Fig.3

Incorporation des bibliothèques CUDA au projet, pour l'édition de liens.

```
print_vector(&v);
run(&v);
print_vector(&v);
return 0;
}
```

Même si l'exemple est simplissime, il est riche en enseignements. Nous commençons par initialiser la librairie comme il se doit. A cette étape nous prenons bien soin de préciser quelle est la GPU qui devrait exécuter les kernels. Nos machines de tests ne comportent chacune qu'une GPU et nos essais montrent que tout fonctionne sans appeler l'API `cudaSetDevice`, dans ce cas. Cependant la documentation est claire. La GPU doit toujours être sélectionnée explicitement. Ensuite nous fabriquons un vecteur dont nous initialisons les trois coordonnées à -1. CUDA, comme c'est le cas de l'immense majorité de ce genre de librairies, définit ses propres types. Ici CUDA fournit un jeu de fonctions préfixées par `make_` pour construire ces types.

Nous entrons maintenant dans la fonction `run` de notre exemple. Nous commençons par allouer de la mémoire dans la GPU via l'API `cudaMalloc`. Celle-ci s'attend à recevoir en argument un pointeur sur un pointeur devant recevoir l'adresse de la mémoire réservée à l'issue de l'appel. D'où le peu sympathique `((void**)&gpu_v` dans le code. Ensuite nous copions les données dans la mémoire réservée, au moyen de l'API `cudaMemcpy`. A nouveau cette fonction travaille avec des pointeurs. Cela est d'ailleurs inévitable, la mémoire cible étant inaccessible à la CPU.

De cela nous déduisons que CUDA sera utilisé avec profit en transférant un volume de données aussi grand que possible à chaque fois. En travaillant avec un seul vecteur, notre exemple est pitoyable sur le plan de l'efficacité :) Quand tout est prêt, nous lançons le kernel. Son code ressemble à du C ordinaire.

On peut même dire que c'en est. Comme nous l'avons dit ce code est exécuté dans un thread. Chaque thread dispose d'une variable intégrée `threadIdx`, qui est une structure de données décrivant le rang du thread dans le bloc. Comme les blocs de threads peuvent comporter jusqu'à trois dimensions, la structure de données `threadIdx` définit trois éléments `x,y,z`.

6 LES KERNELS CUDA SONT ASYNCHRONES

Revenons sur le lancement de notre kernel:

```
incVec<<<1, 1>>>(gpu_v);
```

La syntaxe spéciale à base de chevrons définit la constitution du bloc, ou même, pour être très rigoureux, de la grille de blocs de threads à instancier. Le premier paramètre dans les chevrons décrit la grille. Ici une grille de dimensions 1 à un élément, donc finalement pas une grille mais un seul bloc. Le deuxième paramètre décrit le bloc. Le 1 signifie un seul thread dans le bloc. Si nous avions :

```
incVec<<<1, 10>>>(gpu_v);
```

Le bloc n'aurait qu'une dimension et comporterait 10 threads. Si nous avions :

```
dim3 dimBlock(3, 3); // CUDA vient avec des types C++ :)
incVec<<<1, dimBlock >>>(gpu_v);
```

Notre bloc serait un "cube" de threads contenant 27 threads. Ceci sera expliqué plus en détail le mois prochain. Revenons à notre exemple. Nous avons un seul thread qui est donc de rang 0. Son `threadIdx.x` vaut zéro. En affectant cette valeur à toutes les coordonnées de notre vecteur, nous avons incrémenté ses coordonnées de 1 :) Il y a encore un point très important à toujours avoir à l'esprit quand on travaille avec les kernels: tout se déroule de manière asynchrone, même si ça ne se voit pas dans le code. Ainsi `incVec<<<1, 1>>>(gpu_v);` rend immédiatement la main au flux d'exécution côté CPU. Comment être alors averti de la fin des opérations côté GPU ? On ne peut pas. En revanche les opérations de transfert de données en mémoire se comportent comme des barrières. Ainsi, lorsque le code côté CPU arrive entre dans

```
cudaMemcpy(v, gpu_v, size, cudaMemcpyDeviceToHost);
```

l'API attend la fin de l'exécution des threads pour transférer les données et récupérer le résultat.

7 ETRE ATTENTIF AUX QUALIFICATIFS DE FONCTIONS

Le terme `__global__` présidant à la déclaration de notre kernel mérite toute notre attention. Selon les qualificatifs utilisés, l'accès au kernel par le code est modifié:

Qualificatif	Exécution	Invocation
<code>__global__</code>	GPU	depuis le côté CPU seulement
<code>__device__</code>	GPU	depuis le côté GPU seulement
<code>__host__</code>	CPU	depuis le côté CPU seulement

On notera encore que `__device__` et `__global__`, autrement dit le côté GPU, ne supporte pas la récursivité, ni les variables statiques. Il existe d'autres limitations que le lecteur découvrira dans la documentation.

8 CONCLUSION PROVISOIRE

Le lecteur trouvera dans les sources un exemple montrant comment travailler en C++ avec CUDA. Le principe est simple: invoquer les kernels à travers un wrapper déclaré extern "C". Et le code C++ susceptible d'être appelé par ces wrappers sera lui aussi déclaré extern "C". CUDA ne nous a pas livré tous ses secrets, bien loin de là. La manipulation des blocs de threads mérite d'être approfondie, la synchronisation des threads dans un même bloc également. Ce que nous ferons dès le mois prochain :)

■ Frédéric Mazué - fmazue@programmez.com

L'information permanente

- L'actu sur Programmez.com : le fil d'info quotidien
- La newsletter hebdo : la synthèse des informations indispensables.

Abonnez-vous, c'est gratuit !

www.programmez.com

Faites parler votre PC avec Microsoft Speech

Le vieux rêve de science-fiction, l'ordinateur qui parle, est devenu une réalité depuis longtemps sur la machine de Monsieur tout le monde. Avec l'API Microsoft Speech nous faisons aujourd'hui parler votre PC. Nous le ferons écouter dans un prochain article.



En 1968 Stanley Kubrick tourne "2001 Odyssée de l'Espace". Le personnage central est l'ordinateur HAL, capable de synthèse et de reconnaissance vocale. A

l'époque, l'ordinateur qui parle est du domaine de l'imaginaire, même si les premières recherches en synthèse vocale datent de 1965. Aujourd'hui la synthèse et la reconnaissance vocale sont présentes dans tous les PC et tout passionné de programmation peut s'amuser à "humaniser" les applications qu'il écrit. Nous abordons aujourd'hui le travail avec l'API Microsoft Speech. Le domaine est vaste mais très amusant, sans être très difficile. Nous commençons par la synthèse vocale. Nous traiterons de la reconnaissance vocale dans un article ultérieur.

1 LES OUTILS

Comme la majorité des API Windows les plus intéressantes, Microsoft Speech est une API COM. Cela signifie qu'on peut la manipuler avec n'importe quel langage à pointeurs. Pascal ou C par exemple. Cependant, la gestion des erreurs et des ressources en programmation COM devient vite extrêmement fastidieuse avec des langages de trop bas niveau comme ceux précités. Nous utiliserons donc C++ et une classe de pointeurs COM intelligents pour une meilleure organisation du code et une plus grande facilité d'écriture. Pour nos exemples nous avons employé Visual Studio 2008, mais n'importe quel compilateur C++, par exemple le compilateur gratuit Mingw, fera l'affaire pourvu qu'une Windows SDK, apportant les fichiers en-têtes (.h) et les bibliothèques (.lib), soit installée sur la machine. Une API COM peut aussi être manipulée par des langages de plus haut niveau qui proposent une interface à cet effet. Ainsi nous verrons brièvement comment procéder avec un langage de la plateforme .Net (C#) et avec Python. Le lecteur pourra sans difficulté transposer les règles données avec des exemples C++ vers son langage 'COM capable' favori :

2 NOTION DE PROGRAMMATION COM

Une API COM expose des interfaces qui sont des contrats au sens de la programmation objet. Tout comme une interface en Java ou en C# est un contrat. Au moyen de quelques APIs génériques, on obtient des instances de ces interfaces. Plus précisément on obtient un pointeur sur ces instances et grâce à ce pointeur il est possible d'invoquer les méthodes de l'interface. C'est au programmeur qu'il incombe de gérer ces ressources que sont les instances d'interface. En interne chaque interface détient un compteur de

référence. Quand celui-ci tombe à zéro le système libère les ressources. Obtenir une interface via une API incrémente automatiquement son compteur. Chaque interface expose les méthodes AddRef et Release pour que le programmeur puisse mettre à jour ce compteur. D'autre part COM préfère généralement et à juste raison l'agrégation à l'héritage. Ainsi les interfaces complexes sont des agrégats d'interfaces plus simples. Chaque interface COM expose une méthode QueryInterface qui permet d'accéder aux éléments constituant l'agrégat. Cet article n'étant pas dédié à la programmation COM nous n'allons pas pousser plus avant ces notions qui sont suffisantes à notre but d'aujourd'hui. En ce qui concerne la gestion des compteurs de références, notre pointeur intelligent, SmartCOMPtr, va s'occuper de tout. Vive C++ et ses templates !)

3 HELLO WORLD

Nous allons à la fois nous plier à la tradition et y déroger. Cette fois, il n'est pas question d'afficher "Hello World!" à l'écran, mais de le faire dire à l'ordinateur :) Et nous allons le dire de plusieurs façons. Normalement, avec une voix plus grave que la normale, de façon synchrone, le programme attendant la fin de la prononciation de la phrase pour poursuivre, de façon asynchrone le programme se poursuivant, et enfin nous prononcerons des phrases résidant dans un fichier. Tout cela avec quelques lignes de C++ seulement (projet Basic dans les sources) :

```
#include "COMUtils.h"
#include <sapi.h>
#include <iostream>

#error Adaptez le code ci-dessous à votre configuration
const WCHAR* chemin = L"c:\\temp\\helloworld.txt";

void Run()
{
    HRESULT hr;
    SmartCOMPtr<ISpVoice> ispvoice;

    hr = ::CoCreateInstance(
        CLSID_SpVoice,
        NULL,
        CLSCTX_ALL,
        IID_ISpVoice,
        reinterpret_cast<void*>(&ispvoice));
    COMHelper::TestOk(hr);

    hr = ispvoice->Speak(L"Hello world", SPF_DEFAULT, NULL);
```

```

COMHelper::TestOk(hr);
std::cout << "Hello World synchrone" << std::endl;

std::cout << "Bemol sur un Hello World synchrone " << std::endl;
hr = ispvoice->Speak(
    L" <pitch middle = '-10' /> Hello world", SPF_IS_XML, NULL );
COMHelper::TestOk(hr);

hr = ispvoice->Speak(
    L"Hello world", SPF_ASYNC|SPF_PURGEBEFORESPEAK, NULL);
COMHelper::TestOk(hr);
std::cout << "Hello World asynchrone" << std::endl;
// Sans temporisation la terminaison du programme
// coupe le sifflet à la voix :)
::Sleep(2000);

std::cout << "Debut d'un long speech" << std::endl;
hr = ispvoice->Speak(
    L"Hello world number one",
    SPF_ASYNC|SPF_PURGEBEFORESPEAK, NULL);
COMHelper::TestOk(hr);
hr = ispvoice->Speak(L"Hello world number two", SPF_ASYNC, NULL);
COMHelper::TestOk(hr);
std::cout << "Attente de la fin du speech" << std::endl;
hr = ispvoice->WaitUntilDone(30000);
COMHelper::TestOk(hr);
std::cout << "Le speech est fini" << std::endl;

std::cout << "Debut d'un long speech depuis un fichier" << std::endl;
hr = ispvoice->Speak(NULL, SPF_PURGEBEFORESPEAK, NULL);
COMHelper::TestOk(hr);
hr = ispvoice->Speak(
    chemin, SPF_ASYNC|SPF_IS_FILENAME|SPF_IS_XML, NULL);
COMHelper::TestOk(hr);
std::cout <<
    "Attente de la fin du speech depuis un fichier" << std::endl;
hr = ispvoice->WaitUntilDone(30000);
COMHelper::TestOk(hr);
std::cout << "Le speech depuis un fichier est fini" << std::endl;
}

int main(int argc, char** argv)
{
    try
    {
        COMInitializer ci;
        Run();
    }
    catch(COMException ce)
    {
        std::cout << ce.raison() << std::endl;
        return EXIT_FAILURE;
    }
    return EXIT_SUCCESS;
}

```

Commentons ce code. La classe COMInitializer se charge d'initialiser COM. Cette opération est requise quelle que soit l'API COM avec laquelle on travaille. Dans le cas de SAPI, il faut en outre inclure l'en-tête

sapi.h et faire une édition de liens avec sapi.lib. L'API Microsoft Speech, ou SAPI pour les intimes, expose une ribambelle d'interfaces COM. En ce qui concerne la synthèse vocale, l'interface principale est ISpVoice et notre code commence par en obtenir une instance au moyen de l'API CoCreateInstance. Ceci fait, il suffit d'invoquer la méthode Speak de cette interface pour entendre notre machine parler. Le texte passé à la méthode doit être en Unicode Windows, ce qui est très classique avec COM. Les flags passés en argument à la méthode méritent toute notre attention. Nous avons utilisé SPF_DEFAULT, ce qui fait parler la machine en mode synchrone et à une hauteur par défaut également. L'appel suivant à Speak fait parler la machine à une hauteur plus basse. Pour cela, le texte à prononcer est :

```
L" <pitch middle = '-10' /> Hello world"
```

La hauteur de la voix est définie dans une balise XML. C'est le flag SPF_IS_XML qui fait que l'API va au préalable analyser le texte à la recherche de balise et se régler en conséquence. Si le flag SPF_IS_XML est omis, le résultat sera assez surprenant :) Le reste de notre exemple montre comment faire parler la machine en mode asynchrone et en lisant le texte depuis un fichier. Tout cela est très simple et n'appelle pas plus de commentaires. Mais attention, comme mentionné dans un petit coin de la MSDN, le nom du fichier doit être complètement qualifié, autrement dit le chemin doit être absolu et complet, sinon l'API échoue.

4 AVEC C#, PYTHON OU D'AUTRES LANGAGES

L'API Speech peut être utilisée avec n'importe quel langage sans pointeur pourvu que celui-ci soit capable de travailler malgré tout avec COM. C'est le cas de tous les langages .Net. Sous Visual Studio c'est très simple. Il suffit d'ajouter la librairie de types COM Microsoft Speech aux références du projet. A la suite de quoi des adaptateurs seront automatiquement générés et la présence de COM est totalement transparente. On invoque les méthodes des interfaces COM comme on le ferait avec des classes normales :

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using SpeechLib;

namespace BasicCSharpSpeech
{
    class Program
    {
        static void Main(string[] args)
        {
            SpVoice voice = new SpVoice();
            SpeechVoiceSpeakFlags flag =
                SpeechVoiceSpeakFlags.SVSFDefault;
            voice.Speak("Hello world", flag);
        }
    }
}

```

Si l'on veut travailler avec un langage de script, il suffit que celui-ci propose un module pour l'interface avec COM. C'est par exemple le

cas de Python. Ainsi nous avons fait des essais fructueux avec Python 2.5 associé au module d'extension comtypes que vous pouvez télécharger à <http://pypi.python.org/pypi/comtypes> ou à <http://sourceforge.net/projects/comtypes/>. Moyennant quoi c'est tout simple:

```
from comtypes.client import CreateObject

voice = CreateObject('sapi.SPVoice')
voice.Speak("Hello world", 0)
```

5 SÉLECTIONNER UNE VOIX

Les exemples de cet article ont été essayés sous Windows Vista. Par défaut ce système ne propose qu'une seule voix, qui répond au doux prénom d'Anna. Si vous installez des outils de synthèse vocale tiers, le panneau de configuration de Windows vous permettra de visualiser quelles voix supplémentaires sont disponibles. Ainsi la capture [Fig.1] montre les voix disponibles après installation de la Microsoft Speech SDK 5.1 sur la machine de votre serveur. Il s'agit d'une vieille SDK et toutes les voix ne sont pas garanties fonctionner correctement sous Vista. [Fig.1] Fort bien, mais comment faire pour énumérer les voix par programmation ? Les informations relatives aux voix sont stockées dans la base de registre mais sous une forme non contractuelle. La façon correcte de procéder est d'énumérer des interfaces COM qui, elles, savent comment retirer les informations de la base de registre. Voici un extrait du code C++ pour énumérer des voix (Vous trouverez le code complet sur le site, exemple DemoEnumVoices) :

```
void Run()
{
    HRESULT hr;
    SmartCOMPtr<ISpObjectTokenCategory> pSpObjectTokenCategory;
    SmartCOMPtr<IEnumSpObjectTokens> pSpEnumTokens;

    hr = ::SpGetCategoryFromId(SPCAT_VOICES, &pSpObjectTokenCategory);
    COMHelper::TestOk(hr);

    hr = pSpObjectTokenCategory->EnumTokens(NULL, NULL, &pSpEnumTokens);
    COMHelper::TestOk(hr);

    ULONG nb;
    hr = pSpEnumTokens->GetCount(&nb);
```

```
COMHelper::TestOk(hr);
std::cout << "Nb de voix sur la machine: " << nb << std::endl;
for(ULONG i=0; i<nb; i++)
{
    SmartCOMPtr<ISpObjectToken> pSpObjectToken;
    WCHAR *id;

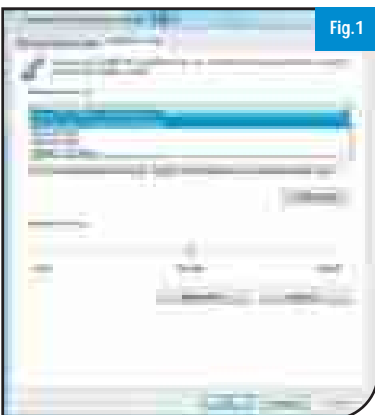
    pSpEnumTokens->Item(i, &pSpObjectToken);
    pSpObjectToken->GetId(&id);
    std::wcout << id << std::endl;
    ::CoTaskMemFree(id);
}
}
```

D'abord une API, SpGetCategoryFromId, donne une interface COM d'énumération d'objets Speech. La valeur SPCAT_VOICES limite le travail de cette API aux voix. Ensuite nous parcourons l'énumération de façon classique. A chaque itération nous passons sur une instance de ISpObjectToken capable de se décrire. Attention à la libération du pointeur COM sur l'identificateur cette interface. Le pointeur est automatiquement obtenu avec la méthode GetId, mais la mémoire doit être libérée manuellement, et avec l'API COM CoTaskMemFree uniquement. Ce genre de facéties est très courant en programmation COM. Le lecteur trouvera encore sur le site l'exemple DemoSelectVoice qui sélectionne la voix Sample TTS présente sur la machine de votre serveur. Cette voix ne connaît que peu de mots et remplace par "blah" tout ce qu'elle ne connaît pas :) L'exemple fonctionnera ou pas sur votre machine selon votre configuration. Ou peut être devrez vous l'adapter pour le faire fonctionner.

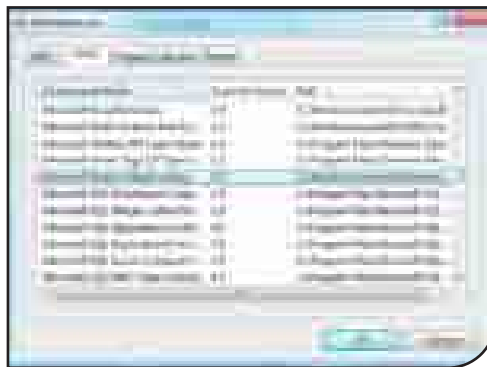
6 SYNCHRONISER L'ACTE ET LA PAROLE

Il peut être intéressant de synchroniser parole et action dans un programme. Plusieurs possibilités sont à notre disposition. Par exemple il est possible de recevoir un message dans une procédure de fenêtre, de manière similaire aux messages WM_PAINT et autres et même de définir une fonction de rappel qui sera automatiquement invoquée à la fin d'un speech. La documentation MSDN donne un code d'exemple complet à ce sujet. Le code est très volumineux (Application GUI en Win 32 brut de décoffrage), et présente peu d'intérêt dans le cadre de cet article. Le lecteur intéressé se reportera à la MSDN. Une autre possibilité, non illustrée dans la MSDN, est bien plus intéressante: il est possible de surveiller le basculement d'un objet de synchronisation Windows, basculement qui se produira à la fin d'un speech asynchrone. L'intérêt est de pou-

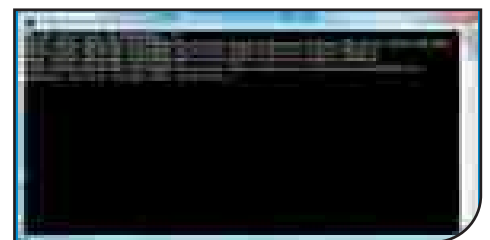
voir surveiller cet événement en même temps que d'autres via l'API Win32 WaitForMultipleObjects. Celle-ci reçoit en argument un tableau d'objets de synchronisation, dont bien sûr un Handle sur une voix, ou un handle sur



Le panneau de configuration sait énumérer les voix présentes sur le système.



Ajout de la librairie de type COM à un projet C# sous Visual Studio



Enumération des voix par programmation

un fichier ou autre. Le travail avec les objets de synchronisation de voix est dévolu à des interfaces agrégées dans ISpVoice. Voici, extrait de DemoEvent sur le site, un code de programmation COM classique qui manipule tout cela :

```
void Run()
{
    HRESULT hr;
    HANDLE event;
    SmartCOMPtr<ISpVoice> ispvoice;
    SmartCOMPtr<ISpNotifySource> pISpNotifySource;
    SmartCOMPtr<ISpEventSource> pISpEventSource;

    hr = ::CoCreateInstance(
        CLSID_SpVoice,
        NULL,
        CLSCTX_ALL,
        IID_ISpVoice,
        reinterpret_cast<void*>(&ispvoice));
    COMHelper::TestOk(hr);

    hr = ispvoice->QueryInterface(IID_ISpNotifySource,
        reinterpret_cast<void*>(&pISpNotifySource));
    COMHelper::TestOk(hr);

    hr = pISpNotifySource->SetNotifyWin32Event();
    COMHelper::TestOk(hr);

    event = pISpNotifySource->GetNotifyEventHandle();
    if(event == INVALID_HANDLE_VALUE)
        COMHelper::TestOk(E_FAIL);

    // Sans l'invocation à SetInterest, le mecanisme
    // de notification reste inactif.
    hr = ispvoice->QueryInterface(IID_ISpEventSource,
        reinterpret_cast<void*>(&pISpEventSource));
    COMHelper::TestOk(hr);
    hr = pISpEventSource->SetInterest(
        SPFEI(SPEI_END_INPUT_STREAM), NULL);
    COMHelper::TestOk(hr);

    std::cout << "Hello World synchrone" << std::endl;
    hr = ispvoice->Speak(L"Hello world with synchronization",
        SPF_ASYNC|SPF_PURGEBEFORESPEAK, NULL);
    COMHelper::TestOk(hr);

    std::cout << "Attente de l'evenement de synchronisation" << std::endl;
    HANDLE tab_handle[1];
    tab_handle[0] = event;
    ::WaitForMultipleObjects(1, tab_handle, FALSE, INFINITE);

    std::cout << "Evenement de synchronisation declenche" << std::endl;
    std::cout << "Le speech est fini :-)" << std::endl;
}
```

Comme mentionné en début d'article, les interfaces agrégées sont obtenues via la méthode standard QueryInterface. Le reste est simple. L'appel à SetNotifyWin32Event provoque la création d'un objet de synchronisation. Un Handle sur celui-ci est ensuite obtenu

avec GetNotifyEventHandle. Le lecteur remarquera ensuite l'appel de la méthode SetInterest de l'interface ISpEventSource; Sans cet appel, le mécanisme ne s'enclenche pas. Le reste est simple, on demande un speech asynchrone, puis on attend le basculement de l'événement avec l'API Win32 sus-mentionnée. Une dernière remarque: ISpVoice, l'interface principale, a des méthodes enveloppe pour les méthodes des interfaces qu'elle agrège, ce qui permet une autre programmation, que le lecteur trouvera en commentaire dans le code sur le CD-Rom.

7 PARLER SANS ACCENT

Faire dire "hello World" à la machine c'est bien, lui faire dire Programmez! c'est mieux :) Seulement voilà, Anna à un accent anglo-saxon épouvantable... Il est possible de s'en tirer malgré tout très honorablement en définissant les mots à prononcer comme une suite de phonèmes, ceux-ci fussent-ils américains. Les phonèmes sont décrits avec leur effets dans la documentation MSDN. Voici un code extrait de DemoPhoneme (sources de l'article) qui prononce Programmez (à peu près) correctement.

```
void Run()
{
    HRESULT hr;
    SmartCOMPtr<ISpVoice> ispvoice;

    hr = ::CoCreateInstance(
        CLSID_SpVoice,
        NULL,
        CLSCTX_ALL,
        IID_ISpVoice,
        reinterpret_cast<void*>(&ispvoice));
    COMHelper::TestOk(hr);

    std::cout << "Programmez anglais :-)" << std::endl;
    hr = ispvoice->Speak(L"programmez", SPF_DEFAULT, NULL);
    COMHelper::TestOk(hr);

    ::Sleep(100);

    std::cout << "Programmez francais :-)" << std::endl;
    hr = ispvoice->Speak(
        L" <PRON SYM = 'p r ao - g r ae - 2 m ey !'/'>", SPF_IS_XML, NULL );
    COMHelper::TestOk(hr);
}
```

Pas de mots à prononcer, mais seulement des phonèmes, à l'intérieur d'une balise XML, comme ceci :

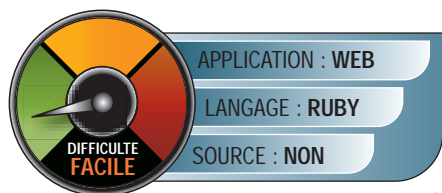
```
<PRON SYM = 'p r ao - g r ae - 2 m ey !'/'>
```

Bien remarquer que les phonèmes sont toujours séparés par des espaces. Le moteur de synthèse vocale perd son latin sans cela :) Donc au final, une syllabe est constituée d'un ou plusieurs phonèmes et les syllabes sont séparées entre elles par un tiret, toujours avec des espaces de part et d'autre. Le chiffre 2 (essayez la valeur 1 aussi) est là pour l'accentuation sur la dernière syllabe. A bientôt pour la reconnaissance vocale.

■ Frédéric Mazué - fmazue@programmez.com

Développez en Ruby on Rails avec 3rdRail

3rdRail est un environnement de développement édité par CodeGear, qui fournit un certain nombre de fonctionnalités afin de programmer des applications Ruby ou – et c’est son but avoué – des applications web avec Ruby on Rails.



L'IDE est basé sur la plateforme Eclipse, ce qui offre donc l'avantage d'un éditeur multi-plate-forme pouvant s'installer sous Windows,

Linux et MacOS X. De plus,

pour les systèmes sur lesquels Ruby et Rails ne seraient pas installés par défaut (notamment Windows), le package d'installation de 3rdRail est livré avec un bundle d'installation comme InstantRails pour Windows. Dans cet article, nous allons aborder les fonctionnalités principales de l'IDE, celles qui permettent dès la prise en main de développer facilement une application simple. Nous réserverons à une future publication les fonctionnalités plus avancées comme le debugging, par exemple.

CONFIGURATION DE 3RDRAIL

Au premier lancement de 3rdRail, il y a principalement une chose à faire : configurer l'interpréteur Ruby à utiliser. Pour cela, rendez vous dans les préférences de 3rdRail, rubrique Ruby / Interpreters. Vous pouvez ici configurer une liste d'interpréteurs avec pour chacun, des arguments ou des bibliothèques externes différentes : [Fig.1]

Ajoutez votre interpréteur en cliquant sur le bouton "Add" et en indiquant les paramètres adéquats dans la fenêtre qui s'ouvre alors : [Fig.2]. Le type d'interpréteur est soit un "Generic Ruby", pour un interpréteur "classique", ou bien "JRuby", pour l'interpréteur Java du langage.

- Entrez dans le nom de l'interpréteur un nom arbitraire qui vous permettra d'identifier vos différentes configurations.
- Dans le champ de l'exécutable, indiquez le chemin complet du binaire de l'interpréteur. 3rdRail détectera alors automatiquement les bibliothèques à utiliser.
- Vous pouvez ajouter à volonté des bibliothèques externes grâce au bouton "Add External libraries..."

Voilà la seule configuration que vous devez impérativement effectuer

avant d'aller plus loin. Pour le reste, faites selon vos habitudes de programmation : les options sont multiples pour personnaliser l'interface, la coloration syntaxique et l'environnement en lui-même.

CRÉATION ET CONFIGURATION DU PROJET

Créez un nouveau projet Rails en passant par le menu "File > New > Rails Project". Une fenêtre de paramétrage s'ouvre : [Fig.3]

Renseignez le nom de votre projet et, éventuellement, un chemin autre que le chemin par défaut.

Si vous avez configuré plusieurs interpréteurs Ruby, vous pouvez choisir celui qui vous convient dans "Ruby Options" ; laissez le choix par défaut si vous n'en avez qu'un.

Dans les "Rails Options", vous pouvez spécifier la version de Rails à utiliser, si toutefois vous en avez plusieurs d'installées. Si vous souhaitez utiliser la dernière version, exécutez la commande *gem install rails* en console et redémarrez 3rdRail si une nouvelle version a été installée. Enfin, choisissez votre type de base de données et si vous souhaitez ou non figer la version de Rails en l'incluant dans le répertoire "vendor/rails" de votre projet.

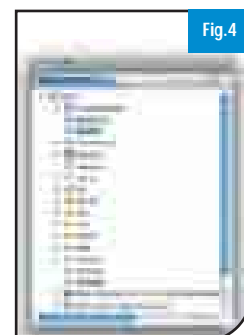
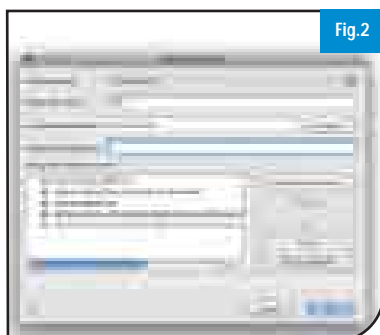
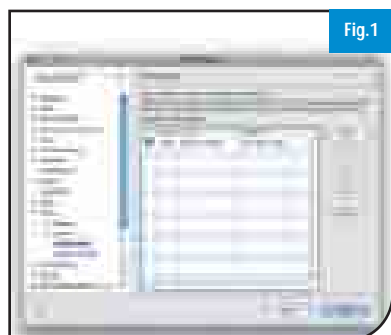
Validez vos choix, et attendez que 3rdRail ait fini de générer l'espace de travail de votre nouveau projet.

RAILS EXPLORER

Une fois prêt, le projet est représenté sous forme d'une arborescence à gauche de la perspective Rails (par défaut) dans un onglet appelé **Rails Explorer**. Cet onglet présente de nombreux avantages comparé au "Navigator" de base d'Eclipse, puisqu'il présente les fichiers du projet non pas selon leur organisation physique comme sur le disque dur, mais de manière **logique**. [Fig.4]

Ainsi, exit le traditionnel dossier "app" et ses sous-dossiers "models", "controllers", "helpers" et "views", ou encore le dossier "db". Vous trouverez à leur place des noeuds comme "Configuration", "Database", "Models" ou bien "Controllers"...

Ces noeuds regroupent différents sous-noeuds, qui sont parfois



redondants dans l'ensemble de l'arborescence, mais qui permettent d'appréhender l'organisation de votre projet de manière beaucoup plus structurée, en retrouvant à sa place chacun des éléments importants de votre application.

Nous reviendrons au fur et à mesure de notre avancement sur l'organisation du Rails Explorer, lorsque nous aurons à naviguer entre différents fichiers.

Pour le moment, ouvrez le fichier `database.yml` qui se trouve sous le noeud "Configuration". Comme vous pouvez le voir, l'assistant de création de projet a bien respecté votre choix de SGBD, et tout est normalement correctement configuré : nom du SGBD, nom de la base de données et socket de connexion à cette base.

Renseignez dans le fichier le nom d'utilisateur et le mot de passe de connexion pour chaque base ; vous pouvez renseigner un serveur différent pour la production, mais nous n'entrerons pas dans les détails de la mise en production dans cet article.

Pour en terminer avec la configuration de ce nouveau projet, créez enfin les bases `blog_development`, `blog_test` et `blog_production`.

UNE PREMIÈRE RESSOURCE

Différents assistants de génération sont disponibles pour créer vos contrôleurs, ressources et modèles. Ils sont disponibles en faisant un clic droit dans le Rails Explorer sur un noeud "Controllers", "Models" ou "Resources" et en choisissant "New Controller", "New Model" ou "New Resource". Voyons avant de continuer les possibilités que ces assistants nous offrent.

Assistant de génération de contrôleur

Nous n'utiliserons pas cet assistant dans cet article car nous nous baserons sur le contrôleur créé pour le *scaffolding* du modèle **Billet**. Il m'a semblé toutefois utile de vous le présenter ici.

Voici donc à quoi ressemble la fenêtre de cet assistant : [Fig.5]

L'utilisation est simple : vous indiquez le nom du contrôleur à créer, puis listez toutes les vues que vous voulez voir générées pour ce contrôleur. Les actions correspondant aux vues seront bien sûr définies dans le fichier ruby du contrôleur.

Assistant de génération de ressource et de modèle

3rdRail associe la création d'une ressource à la création d'un modèle, et les fenêtres d'assistant sont donc très similaires. A dire vrai, il n'existe qu'une différence : l'assistant de création de ressource propose la génération automatique d'un *scaffold* pour la ressource, qui n'est pas disponible dans l'assistant de génération de modèle.

Nous ne verrons donc que l'assistant de génération de ressource, qui nous intéresse par ailleurs pour continuer notre blog. Créons

donc une ressource **Billet** avec cet assistant : [Fig.6]

Donnez au modèle le nom "Billet", puis créez deux attributs :

- **titre**, de type **string**
- **texte**, de type **text**

Comme vous pouvez le voir, les colonnes du tableau d'attributs incluent d'autres paramètres que le nom et le type de l'attribut. Il s'agit ici des paramètres de validation de vos attributs, qui vous généreront automatiquement les lignes telles que `validates_presence_of :attribut` (pour la validation "required") dans le fichier ruby du modèle.

Tous les types de validation sont ici paramétrables, il suffit pour cela de cliquer dans la colonne adéquate pour l'attribut choisi, puis de cliquer sur le bouton qui apparaît alors. Suivant le type de validation, la fenêtre ouverte comportera plus ou moins d'options.

Ajoutons une validation "required" sur nos deux attributs : [Fig.7]

Vous pouvez alors choisir lors de quel événement la validation sera vérifiée (création, sauvegarde, mise à jour du modèle), et indiquer un message d'erreur et des conditions de validation.

Le fonctionnement des autres fenêtres de validation est similaire. Ajoutez une validation "required" pour nos deux paramètres, et explorez éventuellement les autres possibilités avant de continuer.

Vous remarquerez en bas de chaque fenêtre une petite icône "?" qui vous donne accès à une aide contextuelle sur la fenêtre et, souvent, des liens vers les documentations officielles de Rails ; n'hésitez pas à l'utiliser !

Revenons à la fenêtre de création de la ressource. En dessous des attributs, vous pouvez choisir d'ajouter automatiquement les attributs `created_at` et `updated_at` classiques, avec la case à cocher "Automatically create timestamp attributes for the generated Model".

Vous pouvez également demander à ce que la migration générée pour le modèle soit automatiquement exécutée.

Enfin, la dernière option permet de générer un scaffold pour la ressource, ce qui créera donc un contrôleur associé et toutes les actions de création, d'édition et de suppression à la mode RESTful. Validez maintenant la création de la ressource. Les fichiers sont générés et vous pouvez voir en bas de l'IDE, dans la console, que des appels sont effectués pour vous, y compris la migration de la base de données.

Pour en savoir plus :

<http://www.codegearfrance.com>

■ Olivier Lance

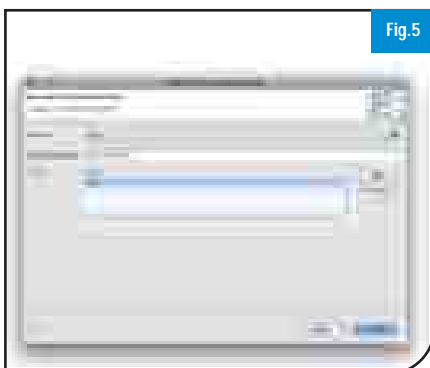


Fig.5



Fig.6

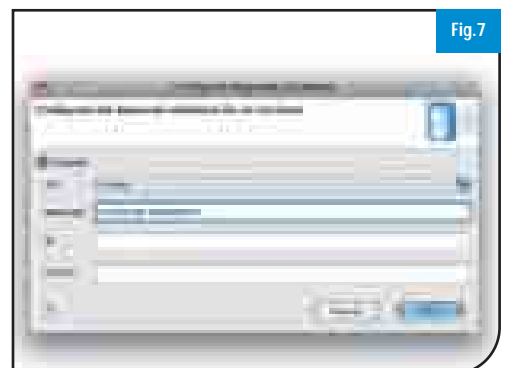


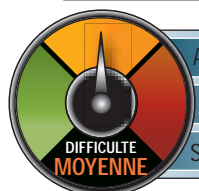
Fig.7

Intel Parallel Studio

La chasse aux bogues parallèles avec Intel Parallel Inspector



Nous avons été formés à programmer en mode sériel et passer à la programmation parallèle n'a rien d'évident, mais plusieurs technologies existent pour vous faciliter la vie : Intel Threading Building Blocks ou le standard OpenMP par exemple.



APPLICATION : MULTICORE

LANGUAGE : C++

SOURCE : NON

Une fois les concepts de base assimilés, implémenter un programme parallèle qui utilise à peu près efficacement un processeur à deux ou quatre cœurs va assez

vite si votre algorithme s'y prête. Mais il est surtout très facile de transformer un logiciel sériel sans bogue en un logiciel multithreadé fortement bogué sans que ce soit évident, ni à détecter ni à corriger. Pour répondre à ce problème, Intel a compris très tôt qu'il fallait aider les développeurs avec des outils logiciels. Pour les développeurs sous Visual Studio en C/C++ cet outil est Intel Parallel Studio, plus spécifiquement Parallel Inspector pour la partie isolation et correction de bogues parallèles que nous allons étudier aujourd'hui.



UN EXEMPLE SIMPLE POUR VISUAL STUDIO

Partons d'un exemple concret : un logiciel qui trouve les nombres premiers dans un intervalle donné. Partons du code parallélisé par OpenMP, après ajout d'une simple ligne de "pragma" au code sériel. Visual studio (2005 ou 2008) doit être installé, puis téléchargez et installez Intel Parallel Studio dont la version bêta est disponible gratuitement en ligne sur intel.com/go/parallel et sur le CD du mois dernier de Programmez !. Créez un nouveau projet nommé "premiers" d'application en ligne de commande et décochez les en-têtes précompilés.

Remplacez le contenu de "premiers.cpp" par le contenu du fichier téléchargé et sauvez, ensuite cliquez droit dans l'explorateur de solution sur "premiers". Dans le menu vous avez un sous-menu "Intel Parallel Composer" en plus, dans lequel vous pouvez choisir d'utiliser le compilateur Intel. Cliquez et observez la nouvelle icône bleue "C++" utilisée pour votre solution. Nous avons un code qui fait appel à OpenMP, mais encore faut-il activer la bonne option de compilation : demandez les propriétés de votre projet, allez dans "C/C++ / Language / General". Vous avez un item OpenMP support que vous devez mettre à /OpenMP. Maintenant, compilez en mode debug et exécutez-le plusieurs fois sur une machine parallèle (Multi-Core, HyperThreading, ...) en surveillant dans le gestionnaire de processus Windows que vous utilisez plusieurs cœurs.

EXEMPLE SIMPLE MAIS BOGUÉ

Le décompte de nombre premiers est différent pour des exécutions successives, ce qui est bien sûr impossible, nous avons donc un bogue. La version sérielle n'avait pas ce problème, il s'agit donc d'un

problème dû à l'exécution parallèle. En rendant notre logiciel parallèle avec OpenMP nous avons probablement omis de protéger certaines parties, mais lesquelles ?

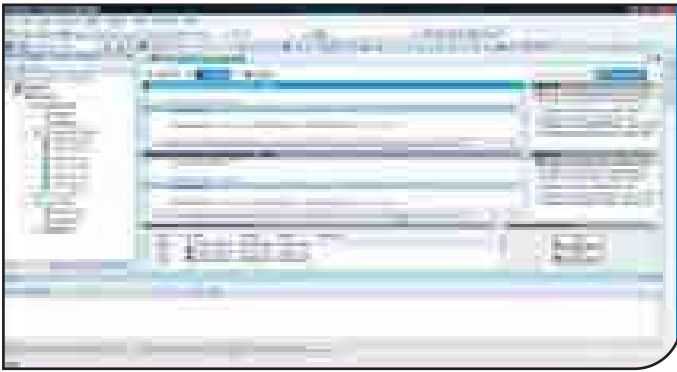
Il est parfois facile comme ici de détecter manuellement un bogue parallèle, dans certaines conditions expérimentales c'est presque impossible. Le bogue parallèle est un animal sournois et dur à chasser, il peut se montrer très discret sur votre station de développement et n'apparaître que sur votre serveur de production.

Intel Parallel Inspector est capable de détecter des problèmes potentiels dans l'exécution parallèle même s'ils n'arrivent pas dans les conditions de votre exécution. Il suffit juste d'exécuter la partie de votre logiciel que vous souhaitez analyser compilée en parallèle dans Parallel Inspector et il trouvera à coup sûr les problèmes dans votre code, magique ! Il faut qu'Inspector ait accès aux symboles pour remonter jusqu'au code source à l'endroit précis du problème, donc compilez en mode debug.

ANALYSE DES ERREURS DE THREADING

Lancez maintenant une analyse directement depuis Visual Studio par le menu "Outils / Intel Parallel Inspector / Check threading errors". Vous pouvez demander différents niveaux d'analyse : Parallel Inspector instrumente les accès mémoire, ce qui est assez lent en soi mais qui ralentira aussi l'exécution. Il faut donc soumettre une charge de travail réduite pour que l'exécution ne dure pas trop longtemps. Par contre cela n'influe pas sur la fiabilité des résultats, nous cherchons ici à détecter des bogues, pas à augmenter ou mesurer la performance.

Au début de l'exécution, vous voyez défiler dans les logs les bibliothèques



utilisées et celles qui sont compilées en debug, elles sont instrumentées, ce qui pour un gros logiciel est une opération intensive, mais qui sera réutilisée pour de futures exécutions.

UN RÉSULTAT CLAIR ET CONCIS RELIÉ À VOTRE SOURCE

Voici le rapport qui apparaît : vous avez une liste des problèmes en haut, P1, P2, P3 qui sont dans les 3 cas des "data race", soit des interactions de différents threads vers les mêmes variables. Pour chaque problème vous avez dans la liste du bas un détail des interactions. En effet, pour un problème donné exécuté par différents threads vous aurez une liste d'interactions entre différents couples de threads. Ici c'est l'exécution de la même ligne 73 du source qui interagit entre différents threads, double-cliquez sur la ligne et vous verrez la vue complète de 2 threads exécutant deux morceaux de code en même temps (ici le même code, mais pas toujours). La ligne 73 est une variable partagée (gProgress) qui est incrémentée, tout simplement. Une incrémentation est une lecture puis une écriture, selon que la lecture dans le thread du haut arrive avant ou après l'écriture du thread du bas, le résultat est différent. Et bien sûr, deux écritures simultanées sont aussi un problème.

C'est donc une erreur de programmation parallèle ! Toute variable partagée entre différents threads et accédée en écriture doit être protégée. Double-cliquez sur le source pour aller dans l'édition du source à la bonne ligne. La solution est déjà dans le code fourni, ligne 72 qu'il suffit de décommenter en laissant le dièse en début de ligne (la solution proposée n'est pas optimale mais fonctionne). Une section critique s'assure que la région protégée (ici la ligne qui suit) ne sera exécutée que par un seul thread à la fois.

Retour au rapport, allez à l'erreur P2 ligne 103 et décommentez la ligne 102 qui est la solution. Puis P3, qui est aussi ligne 102. Ce n'est pas un bogue de l'outil, mais simplement que nous avons deux

bogues bien distincts sur la même ligne, une concerne le tableau, l'autre la variable incrémentée. L'outil raisonne en termes d'accès mémoire, de variables et vous indique les lignes fautives.

Sauvez le source, fermez le rapport et recompilez. Vous pouvez maintenant relancer l'analyse d'erreurs de threads. Un bogue peut en cacher un autre. Cette fois nous avons un bogue un peu plus complexe, une interaction entre 2 threads mais qui exécutent un morceau différent du code, lignes 75 et 73. La ligne 75 lit la variable gProgress qui est incrémentée juste avant, ligne 73. L'incrémentation en elle-même est maintenant protégée grâce à la section critique, mais si le thread du bas incrémente la variable alors que le thread du haut passe de la ligne 73 à 75 nous avons un bogue. Il faut donc s'assurer que tout le bloc 73 à 78 est exécuté par un seul thread à la fois d'une seule traite, en pratique étendre la section critique avec des accolades, ouverte ligne 73 et fermée ligne 78. Recompilons et exécutons à nouveau dans Parallel Inspector: cette fois 0 erreur ! Nous avons maintenant un logiciel parallèle sans bogues de parallélisme !

LES LIBRAIRIES AUSSI !

Vous avez corrigé les bogues que vous avez créés, mais en tant que développeur vous êtes responsables de tout ce qui est appelé par votre application. En effet, toute librairie lancée depuis une région parallèle doit aussi être garantie sans bogues parallèles. Bonne nouvelle, Parallel Inspector trouvera aussi ces bogues, par contre, faute de version debug de ces librairies et du code source vous ne pourrez les corriger, mais au moins vous saurez qu'il ne faut pas utiliser ces librairies en parallèle.

CONCLUSION

Vous avez pu voir dans cet exemple que les bogues parallèles sont complexes à isoler et corriger mais qu'Intel Parallel Inspector facilite et accélère grandement le travail. Il ne se substitue pas à l'apprentissage des concepts de base du parallélisme, mais vous accompagne dans votre projet en garantissant la qualité du code et des librairies.

Si nous exécutons plusieurs fois le nouveau binaire, le résultat est cette fois le même, ce qui est bon signe, mais le temps d'exécution



est plus long, C'est une des vérités de la programmation parallèle, un logiciel bogué va souvent plus vite ! Mais ne vous inquiétez pas, pour augmenter la vitesse, nous explorerons Intel Parallel Amplifier très bientôt.

■ Paul Guermonprez
Ingénieur logiciel – Intel.

Abonnez-vous à **e-PROGRAMMEZ!**

2,7 € seulement par numéro au format PDF (pour le monde entier)

et pour **0,84 €** de plus par mois : abonnement illimité aux archives
(numéros du magazine, et articles, en PDF)

Abonnez-vous sur www.programmez.com

Créer un jeu vidéo de A à Z avec XNA

2^e partie

Le mois dernier, nous vous avons jeté les bases de la conception d'un jeu de gestion de type Sim City et présenté les parties principales de notre moteur C# / XNA. Pour le moment, le joueur peut construire des bâtiments sur la carte avec les touches clavier. Nous aborderons dans ce numéro la partie statistique de ce type de jeu, de manière à ce que nos villes puissent évoluer et avoir de nouvelles demandes.

La classe mère !

Nous avons rajouté des classes utilitaires dans notre moteur. Notre classe Variable représente une ... variable. Chaque variable peut avoir des dépendances ainsi qu'un poids et un coefficient pour ajuster des calculs. Par définition, une Variable est également une moyenne des dépendances des autres variables. Il est possible de changer son comportement du mode de calcul de ses dépendances (somme simple ou incrémentale). On peut également affecter des événements pré et post-calcul via les méthodes BeforeCompute & AfterCompute. La classe Operation permet de surcharger les opérateurs communs. Elle fait exactement la même chose entre une variable de notre propre type et une variable de numérique type double, int, etc. Enfin, la classe utilitaire GenericHandlers permet un meilleur contrôle des résultats (majoration, minoration, etc.).

A vos marques, prêt !

C'est bien joli tout cela mais il faut maintenant que notre système soit interactif. Pour cela, nous avons créé une classe Simulation. Celle-ci contient un Dictionnaire de List associé à un string. Cela nous permet de définir un " pool " de variables dont on pourra lancer le calcul seulement lors d'un événement spécifique : à la fin de chaque journée ou chaque mois par exemple.

```
public void Simulate(string pool)
{
    // Définit toutes les variables du groupe comme obsolètes
    for (int i = 0; i < this.pools[pool].Count; i++)
        this.pools[pool][i].UpToDate = false;
    // Puis les recalcule
    for (int i = 0; i < this.pools[pool].Count; i++)
        this.pools[pool][i].Compute();

    // Appelle l'événement OnBeforeFinished
    if (this.OnBeforeFinished != null)
        this.OnBeforeFinished(pool, this);

    // Appelle l'événement OnFinished
    if (this.OnFinished != null)
        this.OnFinished(pool, this);
}
```

Je n'aurai pas le temps

Dans le même principe que le dernier numéro, nous allons changer les " vitesses " (GameSpeed) du jeu à l'aide des touches claviers. Nous créons ensuite dans notre GameController la méthode Upda-

teLogic qui permet d'avancer le temps ingame et donc ... d'appeler notre méthode Simulate permettant de calculer toutes les variables du pool défini.

```
public void UpdateLogic(GameTime gameTime)
{
    float dt = gameTime.ElapsedGameTime.Milliseconds / 1000.0f;
    this.totalTime += dt;

    // Si on est pas en pause...
    if (this.speed != GameSpeed.Pause)
    {
        // On a passé un jour
        this.dayCounter += dt;
        if (this.dayCounter >= Rules.SecondsPerDay / (int)this.speed)
        {
            int lastMonth = this.gameDate.Month;
            int lastYear = this.gameDate.Year;

            this.gameDays++;
            this.gameDate = this.gameDate.AddDays(1.0);
            this.dayCounter -= Rules.SecondsPerDay / (int)this.speed;

            // Lance la simulation quotidienne
            this.simulation.Simulate(SimulationPool.Daily.ToString());

            // Appelle l'événement lorsque l'on change de jour
            if (this.OnNewDay != null)
                this.OnNewDay();
        }
    }
}
```

Evolution de la ville

Passons maintenant à un peu de pratique et implémentons côté jeu un système de niveau. On va partir sur quelque chose de générique car d'autres éléments que les villes pourront être amenés à évoluer (comme des bâtiments par exemple). On vous laisse regarder le code de la classe abstraite Level & Levelable disponible sur le site. La classe Level implémente principalement les méthodes LevelUp et CheckLevel. Levelable contient une liste de Level. On va faire hériter notre classe City de Levelable et déclarer notre premier niveau à l'initialisation.

```
// Crée les levels
this.AddLevel(new CityLevel1(this));
```

```
this.AddLevel(new CityLevel2(this));
this.AddLevel(new CityLevel3(this));
```

Notre classe CityLevel contient 3 méthodes importantes : FirstCityUp, CityUp() et CheckConditions(). La méthode FirstCityUp va nous permettre de " construire " nos domaines et CityUp de les enregistrer dans le " pool " afin que leurs variables puissent être mises à jour. Dans la classe city, nous allons définir 3 " Variables " :

```
// Niveau actuel de la ville
private Variable levelVar = new Variable(0);
// Réputation actuelle de la ville.
private Variable reputation = new Variable(0);
// Population actuelle de la ville. Les besoins de la ville
sont dépendants du nombre d'habitants
private Variable population = new Variable(0, new SumVariable
Computer(VariableComputeMode.IncrementalSum));
```

On a regroupé tous les besoins d'une ville (Nourriture, Eau, etc.) dans une classe nommé Domain. Celle-ci contient donc les Variables liées à la gestion des villes (demande, offre de tous les bâtiments d'une ville, stock, etc.) et gère leur dépendance.

```
protected override void FirstCityUp()
{
    this.owner.WaterDomain.ConstructDomain(this.owner, Building
Category.Water, this.owner.Population * 0.007, VariableCompute
Mode.IncrementalSum);
    this.owner.FoodDomain.ConstructDomain(this.owner, Building
Category.Food, this.owner.Population * 0.006, VariableCompute
Mode.IncrementalSum);
}
```

Ces domaines sont ensuite enregistrés dans le pool de simulation grâce à la méthode CityUp() que nous avons vue précédemment.

```
protected override void CityUp()
{
    // Besoins
    this.owner.WaterDomain.RegisterDomain();
    this.owner.FoodDomain.RegisterDomain();
    // Réputation
    this.owner.Reputation.Depend("waterRatio", this.owner.Water
Domain.Ratio, 144, 1.0);
    this.owner.Reputation.Depend("foodRatio", this.owner.Food
Domain.Ratio, 144, 1.0);
    GameController.Instance.RegisterVariable(SimulationPool.
Daily, this.owner.Reputation);
}
```

On a défini que la " réputation " de la ville est dépendante de la satisfaction des besoins d'eau et de nourriture. Il suffit maintenant de déterminer les conditions d'évolution de chaque niveau.

```
protected override bool CheckConditions()
{
    return (this.owner.Reputation.Value >= this.ReputationNeeded);
}
```

Je ne crois que ce que je vois

Notre système est désormais fonctionnel ! Bien évidemment, il serait intéressant que le joueur voit véritablement que sa ville passe au niveau supérieur. On peut tout bêtement appliquer un effet de clignotement sur la ville : durant quelques secondes, on bascule de plus en plus vite entre l'image représentant la ville actuelle et celle du niveau supérieur.

Classe City - Méthode Render

```
if (this.isEvolution)
{
    if(this.flipEvolution)
        recText = new Rectangle((this.Level - 2) * 128, this.
textureOffset * 128, 128, 128);
    else
        recText = new Rectangle((this.Level - 1) * 128, this.
textureOffset * 128, 128, 128);

    this.timeflip += Kernel.Instance.Time.ElapsedGameTime.Milli
seconds;
    this.totaltimeflip += Kernel.Instance.Time.ElapsedGameTime.Milli
seconds;
    if (timeflip >= this.limitTimeFlip)
    {
        this.flipEvolution = !flipEvolution;
        timeflip = 0;
        this.limitTimeFlip -= 15;
        if (this.limitTimeFlip < 45)
            this.limitTimeFlip = 45;
    }
    if (this.totaltimeflip > 3500)
    {
        this.isEvolution = false;
        this.totaltimeflip = 0;
        this.limitTimeFlip = 250;
    }
}
else
    recText = new Rectangle((this.Level - 1) * 128, this.texture
Offset * 128, 128, 128);
```

Et voilà le travail ! On va conclure cette chronique par une petite anecdote technique. Bien souvent, toutes les textures d'un jeu sont regroupées dans des " planches ", généralement dont les dimensions correspondent à un multiple de 2 (par exemple 512 par 512, 1024 par 1024). Les dimensions en hauteur et largeur n'ont pas l'obligation d'être identiques : 256x512, etc. Cette résolution était encore indispensable il y a une dizaine d'années pour des raisons de compatibilité matérielle. Si aujourd'hui ce problème n'est plus à l'ordre du jour, ce procédé est encore souvent utilisé car il apporte un réel gain de performance. Surtout, n'hésitez pas à " améliorer " Programmez Tycoon. Les possibilités de création sont illimitées : nouveau niveau, gestion des stocks, gestion de la pollution, sous-domaine, effet bonus/malus, arrivée et départ de la population, etc. On vous donne maintenant rendez-vous le mois prochain !

CODE SOURCE COMPLET SUR LE SITE DE PROGRAMMEZ !

■ Frédéric PEDRO & Anthony CHEN KUANG PIAO

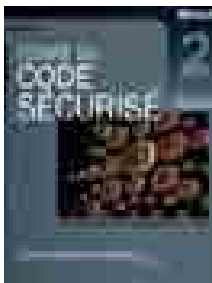
Lauréat de l'Imagine CUP dans la catégorie jeu vidéo et soutenue par Microsoft, ECOThink développe des jeux vidéo innovants grâce à la technologie XNA.
contact@ecological-tycoon.com



Hacking sur le web 2.0

Difficulté : ***
 Editeur : Pearson
 Auteur : collectif
 Prix : 28 €

Nous oublions souvent que l'évolution du web avec les applications et technologies dites " web 2.0 " ont provoqué, certes, de nouveaux usages et des interfaces plus sympathiques mais n'ont guère amélioré la sécurité. Or ce point est souvent passé sous silence et il y a de quoi ! Debugger et sécuriser une application web 2 n'est vraiment pas une partie de plaisir. Bien entendu nous retrouvons les classiques failles : injections de code, exploitation des failles des navigateurs, etc. Mais là où cela se complique, c'est quand on commence à s'attaquer à Ajax et Javascript ou même à une application Flash. Ce livre vous propose toute une série d'attaques avec des contre-mesures possibles. Si vous êtes développeur web, ce livre est **INDISPENSABLE**.

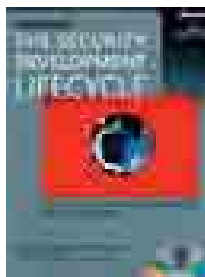


Ecrire du code sécurisé

2e édition
 Difficulté : ****
 Editeur : Microsoft Press
 Auteur : M. Howard & D. LeBlanc
 Prix : 49 €

Ce pavé est l'un des grands classiques pour mieux coder, il nous revient dans sa 2e édition. Celle-ci rajoute de nombreux éléments : comment prendre en compte le codage sécurisé dès le début, prévenir le *evil input* ou le *cross-site scripting* (abordé dans ce numéro de Programmez !), développer en .NET en mode sécurisé, auditer un code sécurisé ou encore les checklists de sécurité pour les développeurs, testeurs et designers... Aujourd'hui encore, peu de développeurs savent créer un code sécurisé. Or, que ce soit pour des applications Web ou desktop, cela est vital. Si ce livre s'oriente naturellement vers le développement Windows /

.NET, on trouve des concepts et modèles de développements applicatifs à d'autres langages et plates-formes. On apprécie dans ce livre l'approche Web et non-Web. La partie Security Testing apporte une autre dimension des tests applicatifs. Les auteurs rappellent aussi quelques méthodes de base et de bons sens mais que l'on aurait tendance à oublier. À la fin de chaque chapitre, on retrouve un petit résumé. L'ouvrage brasse l'ensemble des problèmes, pose les tenants et aboutissants et apporte la parade avec des exemples de code. Franchement, un excellent ouvrage, peut-être difficile à digérer car très (trop ?) dense, mais indispensable pour étudiants et développeurs sérieux.



The Security Development Lifecycle

Difficulté : ***
 Editeur : Microsoft Press
 Auteur : collectif
 Prix : N.C

Les auteurs décomposent SDL en 13 étapes.

Ce n'est pas une théorisation de plus mais des bonnes pratiques réelles, applicables sur le terrain. SDL a l'ambition de réduire les vulnérabilités, les failles dans les projets.

Dès les premières pages, les auteurs sensibilisent les lecteurs sur l'impossibilité, avec les modèles et outils actuels, de fournir du code, des applications sécurisées et au fait qu'il ne faut pas confondre les méthodes de type CMMi et méthodes agiles avec le SDL, elles n'ont pas pour but de fournir une sécurisation du code.

La réussite du SDL dépend du soutien des responsables, de la compétence des personnes la mettant en œuvre, et de la bonne éducation des développeurs et des intervenants aux projets.

Il y a une structure à mettre en place pour garantir la réussite du SDL dans les équipes et les projets. Passionnant et instructif.

De nombreuses références et sources complètent le texte. Livré avec un CD. En Anglais.



Chasser les failles de sécurité

Difficulté : ***
 Editeur : Microsoft Press
 Auteur : collectif
 Prix : 49 €

Une des plaies de la sécurité est l'existence de failles dans les applications. Le codage sécurisé devient vital dans bien des projets mais encore faut-il être sensibilisé et savoir comment les découvrir, les analyser et les corriger. L'approche des auteurs est progressive. Tout passe par le test mais comment les faire ? On pourra passer par une modélisation des menaces. On se mettra successivement en qualité de client malveillant et de serveur malveillant. Puis c'est la longue liste des attaques : overflow, attaques de scripts, problèmes XML, les attaques DoS, les problèmes avec les codes managés de type .Net. Aucune révélation mais le rappel des bonnes pratiques et des menaces bien réelles que tout développeur devrait connaître et régler ! Un ouvrage de qualité, précis. Indispensable malgré un prix un peu élevé !

Techniques de Hacking



Difficulté : ****
 Editeur : Pearson
 Auteur : Jon Erickson
 Prix : 28 €

Rien ne vaut les grands classiques du hack ! L'auteur

de cet ouvrage vous le prouve avec une " facilité " déconcertante. Tout d'abord, on revient sur les techniques de base de la programmation puis on débute notre découverte avec les classiques du genre : débordement de tampon, les chaînes de format. La suite applique des techniques plus " lourdes " comme les attaques réseaux via les sockets, renifler un réseau, scanner les ports, etc. L'auteur propose surtout toute une série de contre-mesures, de la plus élémentaire à la plus avancée et aborde assez longuement la cryptographie.

solutions
linux
opensource

Le Salon européen dédié à Linux et aux Logiciels Libres

31 mars, 1^{er} et 2 avril 2009
Paris Expo - Porte de Versailles



10 ans !

pour visiter le salon et obtenir votre badge d'accès gratuit,
connectez-vous sur www.solutionslinux.fr

un événement


Tarsus

Silver sponsor

 **CANONICAL**

Solutions Linux/Open Source - 2/6 rue des Bourets - 92150 Suresnes
Tél : 33 (0) 1 41 18 63 33 - Fax : 33 (0) 1 41 18 60 68 - www.solutionslinux.fr

Le Savoir est Pouvoir

Nouveau! NetAdvantage for Silverlight Data Visualization
Rendre les projets de Business Intelligence possibles pour
chaque entreprise



Composants pour interfaces utilisateur pour

- Concevoir des tableaux de bord riches
- Visionner les bases de données métiers
- *Donner du pouvoir* aux décideurs

Pour de plus amples infos : infragistics.com

[Sales-europe@infragistics.com](mailto:sales-europe@infragistics.com)

 **N° Vert** 0800 667 307



NetAdvantage®
Data Visualization Silverlight

mapping

charts

gauges

zoombar

timeline

& more!