

Sun rachète MySQL !
Quelles conséquences ?

Le futur de Java

Au cœur des Bases de données

*SQL Server 2008 : offensive DBA
Les nouveautés SQL d'Oracle 11g
Choisir sa base
Ingres et la réplication*

Microsoft 2008

Tout nouveau tout beau !

- PowerShell 2.0
- Étendre et personnaliser Visual Studio 2008
- Volta : l'alternative à Google GWT ?
- Synchronisation facile avec Sync framework

Messagerie
Découvrir
la plate-forme
ejabberd

Architecture
C'est quoi
le DDD ?

Java
Utiliser
l'API Preferences

Ant
Phing :
du build
pour PHP

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

M 04319 - 105 - F: 5,95 €



**DÉVELOPPEZ
10 FOIS PLUS VITE**

WINDEV

WEBDEV

WINDEV
Mobile

PLATEFORME INTÉGRÉE DE DÉVELOPPEMENT

**WINDEV®
XII**



PCSOFT

PLATEFORME PROFESSIONNELLE
DE DÉVELOPPEMENT (AGL)

**Parmi les 500
nouveauautés:**

Accès natif à **SAP**

Fond de page **PDF**

Débogage à **distance**

Compilation «JITc»

Gestion des exigences

Fonctions d'administration
Réseau **SNMP**

Réplication automatique

Sauvegarde à chaud

Héritage de modèle

Nouveau RAD

100 Nouvelles fonctions
Java

39 Nouvelles fonctions
PHP

50 Nouvelles fonctions
Linux

Débogueur **PHP**

Web 2.0/Ajax

Et vous, connaissez-vous
WINDEV 12 ?
Ici présentation de WINDEV
à Paris, une des 12 villes du
Tour de France WINDEV.

UN CODE UNIQUE :

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

Des applications
superbes sans compé-
tence graphique grâce
aux gabarits fournis.
Ergonomie assurée par le
correcteur d'IHM intégré.



112 pages de
témoignages,
sur simple demande

Demandez le dossier technique gratuit (en couleurs, en français), accompagné
de 112 pages de témoignages et d'un DVD. **Version Express Gratuite.**

Tél Province **04.67.032.032** Tél Paris **01.48.01.48.88** info@pcsoft.fr

www.pcsoft.fr

Fournisseur Officiel de la Préparation Olympique

XII

> Actus

L'actualité en bref	6
Agenda	6

> Événements

Sun - MySQL : quelles conséquences ?	8
2008-2009 : où en est Java	12

> Outils

PHING : du ANT pour votre code PHP	16
--	----

> Projets

Comprendre le Domain Driven Design	21
--	----

> Gros Plan

Microsoft 2008 : tout pour le développeur !	24
Comment et pourquoi migrer les projets.Net vers .Net 3.5	28
La gamme Visual Studio 2008 Express	29
Toutes les techniques d'extensibilité de Visual Studio 2008	32
Quoi de neuf dans PowerShell 2.0 ?	36
Sync Framework : tout pour les données	38
Volta : un projet multifonctions	43

> Dossier : SGDB

La base de données au cœur des applications	46
SGDB 2008 : évolution, continuité, rachat	47
Comment choisir sa base de données ?	48
Oracle 11g : les nouveautés SQL et PL/SQL	52
Le renouveau de Ingres	56
SQL Server 2008 : du développeur au DBA	59

> Développement Web

Une galerie de photos sur le web avec HOP (2e partie)	64
RIA pour téléphone mobile (2e partie)	67

> Code

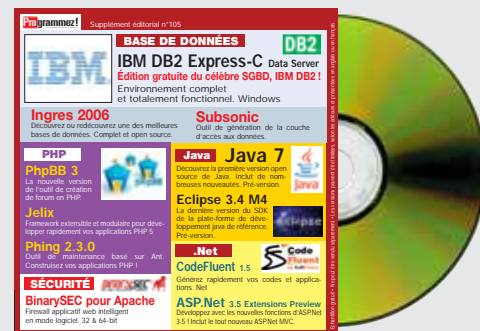
Démarrer avec ejabberd	70
Maîtriser l'API Préférences de Java	73
Introduction à la programmation fonctionnelle avec Haskell (2e partie)	76

> Temps libre

Ludique	80
Les livres du mois	82

Donnez votre avis sur ce numéro

www.programmez.com/magazine_satisfaction.php



CD-Rom 105 PROGRAMMEZ !

IBM DB2 Express-C Data Server

Édition gratuite du célèbre SGDB, IBM DB2 ! Environnement complet et totalement fonctionnel. Windows

Ingres 2006

Découvrez ou redécouvrez une des meilleures bases de données. Complet et open source. Inclut les release 3 et pré-release 4.

Subsonic

Outil de génération de la couche d'accès aux données.

PHP

Jelix

Framework extensible et modulaire pour développer rapidement vos applications PHP 5

PhpBB 3

La nouvelle version de l'outil de création de forum en PHP. Nouvelle interface, nouveaux modules d'administration.

Phing 2.3.0

Découvrez l'outil de maintenance basé sur Ant. Construisez vos applications PHP !

Java

Java 7

Découvrez la première version open source de Java : Java 7. Inclut de nombreuses nouveautés. Pré-version.

Eclipse 3.4 M4

La dernière version du SDK de la plate-forme de développement Java de référence. Pré-version.

.Net

ASP.Net 3.5 Extensions Preview

Développez avec les nouvelles fonctions d'ASP.Net 3.5 ! Inclut le tout nouveau ASP.Net MVC.

CodeFluent 1.5

Générez rapidement vos codes et applications .Net

Sécurité

BinarySEC pour Apache

Firewall applicatif web intelligent en mode logiciel. Il est capable d'apprendre le trafic légitime de toute application ou site web sur Linux.

Drag & Drop Data Conversion

Découvrez Altova MapForce® 2008, le célèbre outil de mappage de données conçu par les créateurs de XMLSpy. Utilisez le glisser-déposer pour mapper, convertir et transformer vos données entre:



XML



Databases



EDI



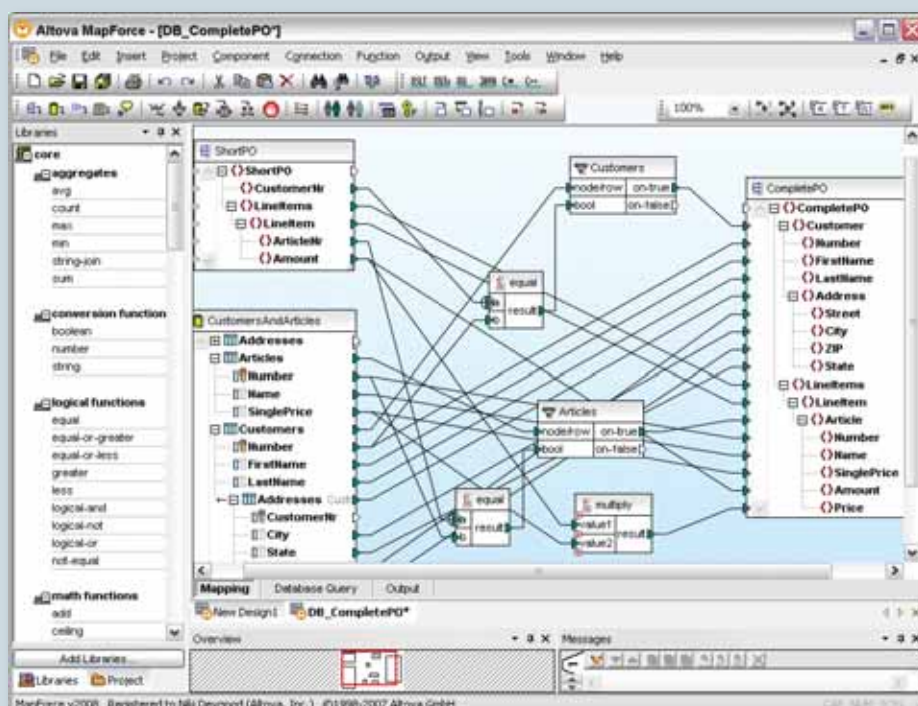
Flat Files



Web Services

- Mapping et conversion de données par glisser-déposer
- Prise en charge des principales bases de données relationnelles
- Fenêtre de requêtes sur bases de données avec éditeur SQL
- Utilitaire FlexText pour l'analyse des fichiers plats
- Prise en charge des messages EDIFACT et X12 EDI
- Connexion des données aux services Web
- Auto-génération de code XSLT 1.0 et 2.0, XQuery, Java, C#, ou C++ pour une utilisation libre de droits
- Création de services Web par glisser-déposer
- Bibliothèque à fonctions extensibles pour le filtrage / traitement des données
- Création de fonctions visuelles pour fonctions personnalisées
- Conversion instantanée des données et fenêtre de sortie

Une fois que vous avez défini un mapping de données dans MapForce, il vous suffit de cliquer sur la fenêtre de sortie pour instantanément convertir les données. Vous pouvez également générer du code libre de droits d'utilisation et le déployer sans frais supplémentaires ou adaptateurs de déploiement. Grâce à MapForce, implémentez l'intégration de vos données et vos applications de type services Web sans avoir à écrire de code!





Réflexions pour 2008

Janvier est souvent le mois des bonnes résolutions que l'on oublie souvent dès le lendemain... Au lieu de faire dans la résolution, je me décide plutôt à faire dans la réflexion, comme quoi l'informatique à haute dose ne détruit, pas encore, tous les neurones...

En préparant ce premier numéro de l'année, nous avons pu dégager quelques tendances, quelques orientations intéressantes et incontournables sur les langages. Loin du fumeux, possible et éventuel Web 3, il y a de véritables changements à attendre dans les langages de développements. On parle tout le temps de langages statiques comme Java, C#. Mais 2008 marquera plus que jamais le raz de marée du langage dynamique. La mouvance web 2 l'avait déjà imposé, aujourd'hui, Sun et Microsoft misent dessus. Sun soutient activement Jruby et Microsoft va lancer dans quelques mois, la DLR pour que .Net supporte directement les langages dynamiques.

Mais au-delà du dynamique, deux tendances se dégagent assez nettement lorsqu'on prend la peine d'observer, d'analyser les évolutions et les recherches. Il s'agit de la programmation fonctionnelle et de la programmation parallèle. Les langages fonctionnels ne sont pas une nouveauté, ils existent depuis des décennies et sont largement utilisés dans les *telcos* ! Or le développeur a tendance à les ignorer superbement. Pourtant, dans le concept de la programmation prédictible, seul le langage fonctionnel permet d'atteindre cet objectif ! Les langages statiques actuels sont trop lourds, trop verbeux pour le faire ! Il est dommage que le fonctionnel n'ait pas suscité d'avantage d'intérêt.

Cependant, les travaux de Microsoft sur le langage fonctionnel F# vont changer à terme la donne. Dans un délai de 3 à 5 ans, pense-t-on. Son intégration à .Net va produire un effet psychologique en fournissant une approche fonctionnelle intégrée. Et pousser d'autres éditeurs, d'autres projets à s'y mettre réellement ! Bref, F# servira de cheval de Troie. Ce sera une réelle révolution dans la manière de voir et de penser la programmation. Souvent, pour qu'une technologie décolle, il faut qu'un grand éditeur l'inclue dans sa panoplie.

Autre chantier, la programmation parallèle. Il y a 2-3 ans, j'étais le premier à mettre en avant le 64-bit, mais force est de constater que le 64 a séduit surtout l'entreprise et principalement le serveur. Alors qu'aujourd'hui le multicore s'impose partout et en premier auprès du grand public, même si peu d'applications exploitent réellement son potentiel. Demain, le développeur pourra utiliser un langage parallèle, appelons-le P#. IBM et Microsoft travaillent à la question. Dès aujourd'hui, les laboratoires réfléchissent à des bibliothèques pour faciliter la parallélisation du code.

Enfin, je conclurai sur la question de Java. Je pourrais presque titrer sur "y'a t'il un pilote dans Java". Sun est en pleine mutation depuis 18 mois. Java appartenant aujourd'hui à la communauté, même si l'éditeur garde la gouvernance, on peut légitimement s'interroger sur son évolution. Malgré les apparences, Java 7 s'annonce presque aussi important que Java 5 dans les changements du langage. Java 5 avait fait un gros travail pour restructurer et simplifier notre bon vieux café. Mais Jruby, ne cache-t-il pas la forêt ? Java continue à chercher sa voie. Trouvera-t-il un nouveau souffle ? Et vous, qu'en pensez-vous ?

■ François Tonic - ftonic@programmez.com

Programmez!

LE MAGAZINE DU DÉVELOPPEMENT

Rédaction : redaction@programmez.com

Directeur de la Rédaction : Jean Kaminsky

Rédacteur en Chef : François Tonic

Ont collaboré : F. Mazué, M. Serrano, C. Queinnec, O. Thery, G. Ponçon, S. Jaber, G. Renard, T. Lebrun, F. Queudret, A. Alcabez, S. Warin, A. Verla, P. Tabassomi, V. Malbete, C. Soutou, S. Padique, C. Pierre de Geyer, I. Van Campenhoudt, J. Sautret, J. Vidames.

Dessin : Michel Piedoue, Jissey

Couverture : © Digital Art/Corbis

Maquette : AJE Conseils

Publicité : Régie publicitaire, K-Now sarl

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

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

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

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

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

PDF : 30 € (Monde Entier) souscription en ligne.

PROCHAIN NUMERO

N°106 - Mars - Parution : 29 février 2008

Model Driven

Révolution ou simple lifting du développement ?

- Comprendre les différentes formes de Model Driven.
- Les outils.

Dossier Open Source

- Gérer un projet Eclipse
- Licence ouverte : comment choisir ?
- L'open source et le collaboratif.

- NE JAMAIS OFFRIR DE CADEAUX JETABLES À SES COLLABORATEURS !

Souriez avec Jissey
www.programmez.com
chaque semaine,
un gif animé



SYSTÈME



Android : évolutions en douceur

Le projet d'une plate-forme open source pour la téléphonie mobile, Android, évolue doucement. Lancée officiellement en novembre 2007, les développeurs et constructeurs disposent d'une version préliminaire. Actuellement, il s'agit de la version m3-rc37a... Cette mouture corrige une série de bugs et apporte de nouvelles fonctions. On notera ainsi : support d'émulateurs multiples (pour le debug dans le Android Debug Bridge), le

plug-in ADT pour Eclipse a bien évolué avec la possibilité de travailler sur les fichiers *manifest*, amélioration de la gestion des skins, support SMS dans l'émulateur. Le grand concours Android Developer Challenge est toujours en cours. Il prendra fin le 3 mars prochain. N'hésitez pas à y participer pour des applications innovantes ! Des podcasts de développement Android sont aussi disponibles pour mieux comprendre le modèle de codage. Les premiers téléphones Android sont toujours attendus pour la mi-2008 ou plus sûrement pour le second semestre.

MOBILE

Le mobile passe au Libre

Si GNU Linux sur téléphone mobile se développe régulièrement, les initiatives éparses n'offraient pas une approche globale. Aujourd'hui, le Linux Phone Standards Forum a publié les spécifications 1.0 qui seront entièrement disponibles en juin prochain. Cette "plate-forme" doit permettre de définir une interopérabilité entre les différentes applications et les services basés sur les mobiles Linux. Il s'agit donc de fournir des spécifications sur un modèle de référence, la téléphonie, le messaging, l'interface et ses services, le carnet d'adresses, les fonctions vocales, etc. Le Forum met en avant l'usage des standards ouverts en prônant l'interopérabilité, qui reste un problème majeur. Le Forum met donc l'accent sur un chemin clair sur le GSM, TCP/IP, WiFi et bien d'autres standards de communication. Et sur les relations avec Android et l'Open Handset Alliance ? Il s'agirait tout d'abord de réduire la fragmentation de Linux Mobile qui n'est pas une bonne chose et nuit à l'interopérabilité et crée la confusion chez les utilisateurs. Un partage de code source, du code commun est envisagé, même si les deux organisations n'ont pas forcément la même approche. Le Forum se veut de plus haut niveau et non ad hoc comme Android, LiMo, Symbian, iPhone, etc. Concernant LiMo, il s'agit d'un autre consortium pour concevoir un système pour téléphonie mobile basé sur GNU Linux. L'éditeur Trolltech a rejoint le consortium début janvier. L'éditeur de Qt va travailler sur les services en ligne, les applications mobiles natives.

WEB

Adobe : objectif Flex 3.0

Rien n'est encore officiel, mais des sources internes prévoient une sortie de Flex 3.0 aux alentours du mois de mars prochain. Actuellement en bêta 3, le travail avance sur tous les fronts. Ainsi, Flex Build Linux est disponible en version alpha depuis mi-décembre 2007 et les extensions

Flex 3 pour Flash CS3 et Dreamweaver sont elles aussi disponibles. Dans le même temps, une bêta de BlazeDS a été dévoilée. Il s'agit d'un serveur Java pour faciliter la distribution des données et le push de celles-ci vers les applications Flex et AIR, à partir du serveur. Initialement disponible dans LiveCycle Data Services ES, BlazeDS est maintenant un projet indépendant en licence LGPL v3.

LIBRAIRIE

Trolltech sur KDE 4 et Windows Mobile !



L'éditeur livrera dans quelques semaines la première version stable de son projet Qt for Windows CE. Disponible en pré-version depuis fin 2007, ce portage s'appuie sur Qt 4.3, supporte Windows Mobile 5 et 6. Il s'agit d'un portage complet des API Qt, sans le support Qt 3 et sans QtOpenGL. On disposera de Qt Designer, Linguist et Assistant. On bénéficie aussi d'une mise à jour de sécurité pour Qt 4.3. Ce *fix* doit résoudre une faille sur QsslSocket qui peut rendre instable la vérification de certificats dans une session SSL. D'autre part, l'éditeur a mis à jour l'Eclipse Integration (v1.0.1). Cela permet aux développeurs Eclipse d'accéder aux API Qt. Enfin, n'oublions pas que Qt a servi de fondation graphique au tout nouveau Kde 4.0, disponible depuis la mi-janvier !

Agenda

FEVRIER

Les 4 et 5 février, Paris 17e Palais des Congrès.

iLearning Forum Paris 2008

www.ilearningforum.eu

Le 5 février au CNIT Paris La Défense

NIDays 2008

Le rendez-vous annuel de l'instrumentation virtuelle
<http://digital.ni.com/>

Le 5 février, Paris 17e, Espace Champerret

Salon emploi des informaticiens ingénieurs IT

www.kavesta.fr

Le 8 février à l'IBM Forum, La Défense, de 9H à 17H

"Rational en Scène"

http://www-05.ibm.com/fr/events/rational_day_a.html

Paris, Palais des Congrès, Porte Maillot

Du lundi 11 février 2008 au mercredi 13 février 2008

Microsoft TechDays 2008

Le rendez-vous incontournable des développeurs et professionnels de l'informatique <http://www.microsoft.com/france/mstd08>

20 et 21 février 2008 - CNIT Paris La Défense

SALON SOLUTIONS INTRANET & TRAVAIL COLLABORATIF

<http://www.groupe-solutions.fr>

MARS

La quatrième Conférence FileMaker francophone se tiendra au Grand Hôtel Aston à Nice du 28 au 30 mars 2008

<http://www.fmconf.com:80/>



Editeur du logiciel BLU AGE™ et propriétaire de la marque Opteams™

Le groupe NETFECTIVE TECHNOLOGY, spécialiste des technologies Objets, UML® et MDA®, propose un catalogue complet de formations et de séminaires. La totalité des sessions proposées est disponible en mode inter et intra entreprise. Retrouvez-nous dans nos locaux à Suresnes (92) ou Bordeaux (33).

Plus d'informations sur www.netfective.com
ou tout simplement contactez-nous au 01 56 05 88 00 / 01 56 05 60 91.

Formations Concepts Objet et UML

- AOO-1j Approche Orientée Objet (1 jour)
4 février 2008; 7 mai 2008; 9 juin 2008; 29 septembre 2008...
- AC-UML-4j Analyse et conception avec UML 2.x, OCL 2.x & MDD (4 jours)
5 février 2008; 13 mai 2008; 10 juin 2008; 30 septembre 2008...
- MD-3j Application d'UML 2.x avec MagicDraw™ (3 jours)
11 février 2008; 16 juin 2008; 13 octobre 2008...

Formations Java, Java EE

- JAVA-4j Programmation avec le langage JAVA (4 jours)
25 mars 2008; 1^{er} décembre 2008...
- J2EE-4j Développement d'applications web pour la plateforme Java EE (4 jours)
26 février 2008; 21 avril 2008; 15 décembre 2008...
- STRUTS-3j Développement d'applications web avec Struts (4 jours)
3 mars 2008; 28 mai 2008; 15 septembre 2008...
- JSF-3j Développement d'applications web avec JSF (3 jours)
6 mars 2008; 2 juin 2008; 18 septembre 2008...
- HIBER-3j Gestion de la persistance avec Hibernate (3 jours)
11 mars 2008; 30 juin 2008; 23 septembre 2008...
- SPRING-3j Développement d'applications avec Spring (3 jours)
17 mars 2008; 15 juillet 2008; 20 octobre 2008...

Formation XML

- XML-CM-3j Technologies XML : Conception et mise en oeuvre (3 jours)
28 avril 2008, 1^{er} septembre 2008...

Formations BLU AGE™

- BLU-INT-1j Présentation synthétique des outils & méthodes MDD™ de BLU AGE™ (1 jour)
15 février 2008; 21 mars 2008; 19 mai 2008; 20 juin 2008; 4 juillet 2008; 5 et 26 septembre 2008; 23 octobre 2008...
- BLU-MDD-5j Une approche pragmatique du MDA : MDD™ avec BLU AGE™ (5 jours)
18 février 2008; 31 mars 2008; 21 mai 2008; 23 juin 2008; 8 septembre 2008; 6 octobre 2008; 24 novembre 2008...
- BLU-SFC-5j BLU AGE™ Software factory configuration (5 jours)
7 avril 2008; 21 juillet 2008...
- BLU-EMD-5j EMDD avec BLU AGE™ - Comment construire des applications complexes avec BLU AGE™ (5 jours)
14 avril 2008; 7 juillet 2008; 25 août 2008; 17 novembre 2008; 8 décembre 2008...

Toutes les marques citées sont la propriété de leurs propriétaires respectifs.



BLU AGE™, le logiciel innovant issu du département de R&D de NETFECTIVE TECHNOLOGY, permet la génération automatique d'applications Java EE et/ou .NET sans recours aux techniques et ressources de développement.

'You design, We generate'

'Vous dessinez, Nous générons'

Découvrez BLU AGE™
Démonstration en ligne de génération d'application.
Agenda : 5, 12 & 19 février 2008 (français) - 7 & 21 février 2008 (anglais).

Inscrivez-vous gratuitement aux prochains webinars sur www.bluage.com.

Formations 2008



Sun – MySQL : quelles conséquences ?



Le 16 janvier dernier, Sun annonçait par communiqué de presse le rachat surprise de l'éditeur open source MySQL pour un milliard de dollars. Une entrée en bourse de MySQL était plus attendue que ce rachat. Il marque un nouvel épisode dans les acquisitions des éditeurs dans les logiciels open source. Mais que signifie ce rachat ? Quelles conséquences ? Comment l'analyser ? Comment le comprendre ?

Tout d'abord, l'arrivée de MySQL dans le giron de Sun propulse ce dernier dans le marché de la base de données évalué à plus de 15 milliards de dollars, un pactole que Sun voudrait bien grignoter. Sun devient ainsi le concurrent frontal de Microsoft, Oracle, IBM, Sybase. Cela confirme plus que jamais l'importance de la donnée dans l'informatique d'aujourd'hui et l'explosion du volume des informations, notamment sur le web, nécessite des bases de données puissantes ainsi que de l'infrastructure matérielle.

Crédibilité de MySQL grâce à Sun !

Soyons honnête, même si MySQL est utilisé par beaucoup d'entreprises dont les plus grandes comme Google, le SGBD devait systématiquement justifier de sa pérennité face à un Oracle, un IBM, un Microsoft, un Sybase. Dans un domaine aussi sensible, les entreprises veulent surtout de la pérennité et joueront souvent la sécurité en misant sur un nom. Et le coût de la licence n'est pas un argument suffisant pour choisir de l'open source. Quand on traite 1 To de données ou des applications très critiques, il faut du solide.

Ce rachat par Sun modifie l'image de MySQL qui trouve d'un coup d'un seul, une crédibilité. Sun apporte désormais à ce SGBD, son nom, sa force, sa pérennité. De quoi s'attaquer à des entre-

prises encore hésitantes. D'autre part, la volonté affichée durant l'annonce de mettre en place des services et supports autour de MySQL, sous le nom de Sun, va là encore rassurer l'entreprise qui cherche un bon support.

Pour Sun, il s'agit aussi d'établir des synergies et opportunités (termes que les dirigeants ont explicitement utilisés) entre MySQL et les logiciels et matériels de Sun. Le côté matériel n'est pas à négliger, car l'éditeur depuis 18 mois a procédé à un virage important en rendant open source ses logiciels, en mettant en place des services et supports et en redevenant un vendeur de matériel. Et là, il y a une forte synergie entre MySQL et les serveurs Sun (bien implantés dans le

haute disponibilité, du cluster, de la montée en charge, des grilles informatiques, etc. Cette compétence peut servir à renforcer les fonctions du SGBD dans ces domaines.

Sun renforce sa place dans l'Open Source

Immédiatement, Sun a revendiqué le M de LAMP (Linux, Apache, MySQL, PHP). Il pourrait aussi le faire avec le M de WAMP ce qu'il ne manquera pas de faire dans les prochains mois, car le marché Windows n'est pas à négliger.

Les nombreux commentaires des éditeurs libres et de la communauté disent que cela crédibilise le modèle économique de l'Open Source, qui devient un choix incontournable. On peut avoir plusieurs avis sur la question. N'oublions pas que MySQL vendait aussi des licences commerciales pour les entreprises ainsi que du support et services (modèle classique du monde ouvert). Sur-

tout, cela ne fait que confirmer la volonté de Sun d'être un acteur de l'Open Source, pouvant peser sur son évolution, les orientations. La mise en open source de ses logiciels depuis 18 mois confirme cela, tout comme celle de Java (première version ouverte avec Java 7). Sun est un acteur impor-

tant avec Open Office, NetBeans, Java, OpenSolaris, Glassfish et maintenant MySQL.

On assiste ainsi à la poursuite d'une certaine concentration des poids lourds de l'Open Source après le rachat de JBoss par Red Hat, de Zend par Citrix ou encore de SuSe par Novell. Qui sera le prochain à passer sous le giron d'un géant du logiciel ? Les concurrents se doivent de réagir rapidement pour ne pas laisser Sun profiter de l'effet de surprise et de la nouvelle aura de MySQL.

Ce rachat fragilise-t-il les autres SGBD ouvertes ?

Pour les uns non, pour les autres oui. Ce rachat peut-il fragiliser des PostgreSQL ou Ingres ? Nous pensons que oui, surtout si Sun renforce les fonctions critiques du SGBD, le mettant à niveau de PostgreSQL, réputé dans ces domaines. Cependant, il est trop tôt pour estimer l'impact de l'annonce. Les prochains mois seront intéressants à suivre. Car MySQL acquiert la puissance de Sun. Notons aussi que Sun a confirmé la continuation du support de PostgreSQL.

Quid des développeurs ?

Difficile de le dire. Rien ne devrait changer sur le court terme. La version 6 de MySQL en cours de développement doit sortir à la fin de l'année. Point d'interrogation sur une éventuelle interaction plus forte entre MySQL et Java

“
**Le branding
(le nom, la marque)
MySQL reste.**”

(Éric Mahé, Sun France)



monde web comme MySQL). Le constructeur propose aussi bien du Solaris que du Linux et du Windows. Donc, pourquoi pas une solution clé en main matériel + logiciel, estampillée MySQL ?

D'autre part, Sun peut apporter à MySQL son expérience de la

Les réactions...

L'avis de Thomas Petazzoni (April)

« Tout d'abord, ce rachat confirme une nouvelle fois la stratégie de Sun en direction du Logiciel Libre : la suite bureautique OpenOffice.org, la machine virtuelle et le langage Java, le système d'exploitation OpenSolaris, le serveur d'applications Glassfish. L'April ne peut que se féliciter d'une telle stratégie, notamment de la part d'une entreprise membre de l'association. D'autre part, ce rachat, pour un montant d'un milliard de dollars, confirme la viabilité du modèle économique de MySQL, basé sur la distribution gratuite de la version libre sous licence GPL de son produit, et sur la commercialisation d'une version sous une autre licence pour les entreprises souhaitant développer des applications propriétaires basées sur MySQL. Ce modèle économique de double licence, également retenu par la société Trolltech pour la bibliothèque Qt, est un des modèles étudiés par le Livre Blanc sur les modèles économiques du Logiciel Libre, publié par l'April au mois de décembre. »

Damien Seguy (Nexen, expert MySQL)

« Beaucoup de surprise avec cette annonce : j'avais fait la prédiction pour 2008 d'une IPO pour MySQL, et cela n'a pas tenu 3 semaines. Ce rapprochement est ce qui ressemble le plus à une IPO, mais ce fut une surprise. Et une bonne. D'un point de vue industriel, le rapprochement fait du sens. Il est difficile pour un éditeur de rester indépendant dans sa gamme de produits. MySQL a sûrement développé la base de données le plus loin possible, et maintenant, il a besoin d'une gamme de logiciels et d'une technologie plus complémentaire pour progresser. Quand on regarde bien, les grands éditeurs en place

offrent bien plus que la seule base de données et ses utilitaires. Ce mouvement de fond me paraît évident dans le monde de l'Open Source, où des spécialités sont maintenant à la recherche de relais de croissance dans les synergies avec d'autres plates-formes. Je rapprocherai ce mouvement de celui de Nexen services en novembre, où nous nous sommes associés à Alter Way pour compléter la gamme de services et technologies offertes. L'avenir est aux éditeurs polyvalents. L'importance de PHP pour MySQL va changer le contexte de ce marché. Certes, le projet PHP reste indépendant comme il l'est actuellement. L'environnement change surtout pour Zend, et je ne serai pas étonné de les voir s'associer avec un grand éditeur à moyen terme. PostgreSQL devrait aussi bénéficier du rachat, et accueillir ceux qui auront eu peur de ce rapprochement, même si cela devrait rester marginal. Dans l'ensemble, ce rapprochement est une bonne opération, et crédibilise considérablement le mouvement Open Source. Les conséquences de ce rachat seront longues à se faire sentir, et les prochains mois vont être cruciaux. D'un point de vue personnel, je suis aussi très heureux pour les fondateurs de MySQL, David Axmark et Monty Widenius. »

Sun : attendre la finalisation

Dans l'effervescence post-annonce, Sun recadre un peu les choses. Tout d'abord, l'éditeur précise qu'il faut attendre la finalisation de l'opération qui doit intervenir d'ici 4-5 mois. « Les activités de MySQL vont continuer en tant que telles. Dans un premier temps il s'agit de fournir des services aux clients » précise Éric Mahé. « Le branding (le nom, la marque) MySQL reste. » poursuit Éric. Dans les prochains mois, on ne devrait pas avoir de grands bouleversements. MySQL continuera à vivre sous son nom. Une fois l'achat terminé et confirmé, on devrait voir apparaître des annonces plus

concrètes. L'ambition est déjà de mieux intégrer MySQL à Java comme c'est le cas avec Postgres (dont le support continuera).

Harold Goldberg (Zend)

« C'est une bonne chose pour l'économie de l'Open Source et du Web. Cela confirme le succès de la pile LAMP comme plate-forme web. La valorisation de la transaction reflète l'adoption de cette solution en entreprise et elle conduit à un accroissement des revenus pour nous. Nous avons une longue expérience de travail avec MySQL et nous sommes encouragés par le rôle des responsables MySQL chez Sun. Ils nous ont dit que nous travaillerons avec Sun comme nous l'avons fait avec MySQL pour faire progresser l'innovation et les standards ouverts. »

Linagora : « beaucoup d'excitation »

« C'est une très bonne nouvelle pour le marché. Cela crédibilise la valorisation (des sociétés open source). Le niveau (élevé) de la revalorisation indique que l'on n'achète pas uniquement du chiffre d'affaires mais aussi la communauté. » précise Alexandre Zapolsky, Président Directeur Général du Groupe Linagora. Pour la SSLL, l'arrivée de Sun est bien perçue, même sur les futurs services que l'éditeur veut mettre en place autour de MySQL. « On a un accord avec MySQL, je crois beaucoup à leurs solutions. Cela va permettre de dynamiser le marché MySQL, ce qui est sympathique pour nous. D'autre part, nous sommes aussi proches de Sun. » poursuit Alexandre. Sur la fragilisation des autres acteurs open source des SGBD, M. Zapolsky ne voit pas de réels problèmes, au contraire, « cela va renforcer les bases open source comme PostgreSQL ou Ingres. MySQL va pouvoir aller sur des marchés où il n'était pas présent et crédibiliser l'alternative à Oracle. »

(Sun travaille activement dans cette voie, reste à en connaître le détail). Bénéficiera-t-on d'une intégration plus fine avec JDBC dont l'utilisation actuelle n'est pas toujours performante. Va-t-il y avoir une intégration MySQL (modules, plugin ?) avec des outils comme Glassfish, Netbeans ? Et quel

impact sur le développement des modules externes ? Oracle peut-il compromettre InnoDB ou IBM la version de SolidDB pour MySQL ?

Et maintenant, il faut attendre

À l'heure où nous écrivons, il reste bien des questions sans

réponses précises comme nous l'avons vu plus haut. D'ici deux mois, nous devrions découvrir les premières offres Sun MySQL sur les services.

Il faudra aussi préciser l'intégration et le rôle des responsables et développeurs MySQL dans Sun. L'opération devrait être finalisée

d'ici juin prochain. Si nous savons déjà que MySQL continuera son activité comme avant et gardera son nom, on connaîtra dans quelques mois le détail de ce que Sun va vouloir faire.

■ François Tonic

ACQUISITION

Oracle se paie Bea !

Surprise. Après des mois de négociations et de revirements, Oracle a finalement accepté de payer au prix fort l'éditeur Bea et va dépenser 8,5 milliards de dollars. Pour Oracle, il s'agissait de mettre la main sur des outils, des technologies dans le middleware, le serveur, la SOA, les applications composites, la sécurité, etc. Les gammes Bea viendront compléter l'offre Fusion et Oracle se dotera d'un solide serveur d'applications. Mais comment Oracle va-t-il intégrer autant d'outils et éviter les doublons logiciels ? Les prochains mois nous le diront. Quid des développements en cours comme Genesis ou l'offre 360 ? Ils devraient continuer, mais là aussi, il faudra attendre la finalisation de l'opération et l'annonce officielle de la stratégie. IBM et Microsoft devront sans aucun doute réagir à cette opération. SAP avait ouvert une nouvelle " guerre " de tranchées avec l'acquisition de Business Object, ou encore SoftwareAG avec Webmethod. Pour le moment, Bea continue à poser les jalons de son futur Genesis, en mettant en son cœur la notion de Dynamic Business Application, avec la SOA, BPM pour les processeurs et le web 2 pour l'interface, l'exposition des applications composites et autres mashup. Le tout s'appuyant sur l'infrastructure Bea.

D'autre part, jusqu'ici taillé pour le marché américain, Bea a décidé de livrer en Europe AquaLogic Enterprise Security 3.0. Parmi les fonctions proposées par l'outil : console d'administration simplifiée des habilitations, de nouveaux modules de sécurité pour les administrateurs, des fonctionnalités de recherche d'attributs configurables pour sécuriser : l'intégration d'applications, les processus métiers et divers environnements applicatifs et données issues de bases de données relationnelles ou d'annuaires LDAP. Il répond à la problématique de la sécurité des services et de la SOA.

SALON

La grand' messe de l'open source et de Linux à Paris pour 3 jours

Comme chaque année, notre tux préféré fait escale à Paris durant 3 jours. L'an dernier, le salon tombait durant le lancement de Windows Vista... La partie conférence, particulièrement riche, aborde des sujets aussi divers que stratégiques : poste de travail, développement Java EE, logiciels libres et temps réel, la plate-forme PHP, la virtualisation, le web 2, la communication IP ou encore SOA dans le monde ouvert... sans compter les tables rondes et les sessions inaugurales.

La partie associative et communautaire occupe cette année encore une large part du salon, pas moins de 500 m2 ! La partie exposition pure couvre 2500 m2 avec 200 expositions ! Parmi les grands éditeurs, on note la présence d'IBM, Red Hat, Ingres, Mandriva, Novell, Mozilla, OpenOffice, Oracle, Talend, MySQL... Parmi les nouveautés présentées, on retiendra : Xwiki Workspaces, un environnement collaboratif, Alfresco (outil de gestion de contenu basé sur Java et proposé par ingenweb), Spago4Q (plate-forme pour assurer la qualité des process de développement au niveau entreprise), Spagic 2.0 (suite logicielle pour concevoir et déployer des infrastructures ESB en relation avec SOA), Quotero (gestion documentaire open source pour entreprise). Compte-rendu complet dans le prochain numéro.

OUTIL

Unify modernise les applications



L'éditeur Unify propose depuis peu Team Developer 5.1, son environnement de développement rapide pour créer des applications d'entreprises. Cette version fonctionne maintenant sous Vista et repose sur Unicode. Cette version

marque la première étape d'un plan de développement très ambitieux de Team Developer. Team Developer 5.1 permet de faire appel à des services Web, ce qui rend les applications Team Developer capables de s'intégrer à d'autres progiciels ou services. L'éditeur propose aussi SQLBase Treasury. Cette édition spéciale de SQLBase crypte les données sensibles. " SQLBase Treasury chiffre les informations de la base de données et protège les applications contre le piratage et les attaques de données personnelles et confidentielles ", souligne Duane George, Vice-président du développement produits et directeur technique d'Unify.

INSTALLATION

RPM en version 5.0

Le package d'installation rpm est désormais disponible en v5 après plusieurs mois de développement. Cette version a été largement réécrite, avec la fin du support des rpmrc. L'outil a été porté sur les principaux systèmes (Windows, Linux, OS X, etc.) et le code a été nettoyé pour gagner en clarté et peut être compilé avec les principaux compilateurs C et C++. On peut maintenant choisir le format rpmdb, la compression est maintenant disponible avec izma. Site : <http://rpm5.org>

COLLABORATION

IBM ouvre la future plate-forme Jazz

Jazz.net, la plate-forme de développement de IBM Rational, s'ouvre à tous les développeurs, sur le modèle d'Eclipse. IBM annonce simultanément IBM Rational Team Concert Express, qui sera l'une des premières "briques" disponibles sur cette plate-forme Jazz, un peu plus tard en 2008 à disposition de la communauté des développeurs. IBM propose IBM Rational Team Concert Express bêta 2, qui intègre des tableaux de bord Web permettant aux équipes de développement de suivre en temps réel le process d'élaboration et le statut des tâches effectuées ou en cours.

OUTIL

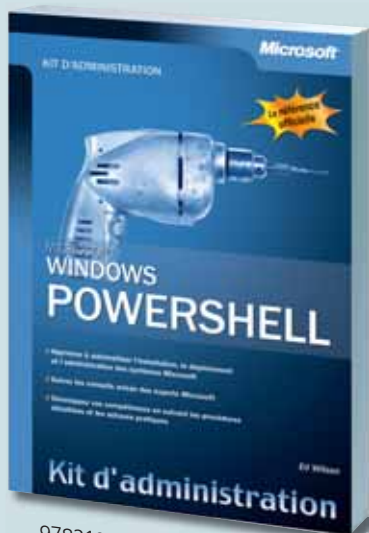
Omnis Studio arrive en version 4.3.1

En novembre dernier, RainingData dévoilait la version 4.3 de son environnement rapide de développement. L'éditeur sortira en mars une mise à jour, la 4.3.1. Cette version apportera des corrections sur l'interface Vista et MacOS X, des améliorations des objets d'interface. La 4.3 apportait un support de données étendues, la compatibilité avec MacOS X 10.5, un développement par équipe amélioré.

Microsoft Press

Préférez l'original !

Kit d'administration



9782100514472 • 336 pages

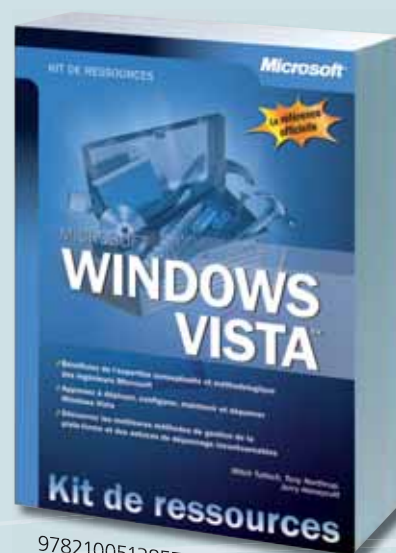
35 €



9782100513840 • 624 pages

49 €

Kit de ressources techniques



9782100513857 • 1 504 pages

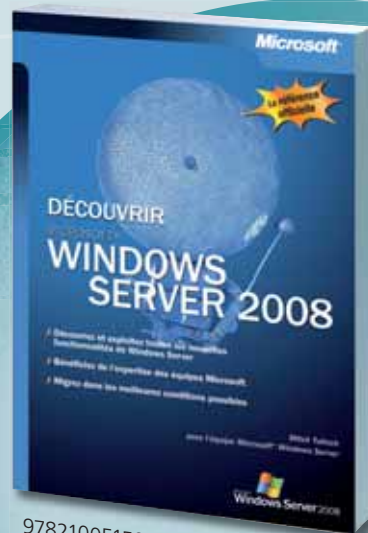
87 €

Les outils du développeur



9782100514496 • 336 pages

29 €



9782100515035 • 504 pages

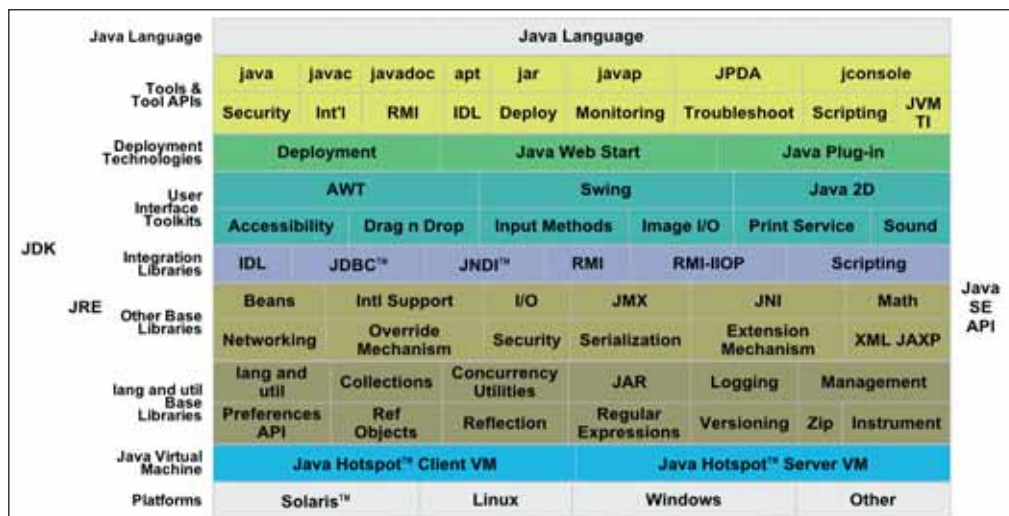
35 €

Découvrez
la nouvelle version
de Windows Server



2008-2009 : Où en est Java ?

Le monde Java continue à évoluer, malgré une activité relativement calme et une communication assez limitée depuis quelques mois. 2008 – 2009 annoncent pourtant de grands chamboulements dignes d'un Java 5 ou d'un Java EE 5.



ressenti depuis quelques semaines, Sun n'organisera pas de Sun Techdays en France comme ce fut le cas en 2007. L'éditeur souhaite toutefois être présent sur différents événements comme dernièrement avec Paris on Rails et le salon Solutions Linux. Cependant, l'éditeur réfléchit à organiser une journée spécifique Java, la JavaDay qui avait eu lieu il y a quelques années. En cas de tenue, elle aurait lieu courant juin. Reste à confirmer la conférence et à définir le contenu et surtout les intervenants. Nous en saurons plus courant février ou mars.

Java 6 Update N : mi-2008

Actuellement en développement, Java 6 Update N introduira un certain nombre de nouveautés importantes pour les applications et développeurs Java. La version finale devrait arriver aux alentours de JavaOne (mai prochain), aucune date précise pour le moment. Cette mise à jour est plus connue sous le nom de code : Consumer JRE. Le Consumer JRE fait partie intégrante de l'Update N et doit apporter à Java une modularité, des performances au démarrage

et en déploiement, jamais vues. Pour faire simple, Update N se concentre sur :

- un nouveau déploiement avec notamment un nouveau toolkit,
- l'apparition du Java Kernel,
- une nouvelle version du Java Plug-In.

Consumer JRE

Le Consumer JRE doit fournir un environnement runtime Java (ou JRE), plus petit, plus modulaire, avec un temps de démarrage réduit, une installation simplifiée, des performances graphiques à la hausse (avec une meilleure intégration de JRE sur le système hôte). Pour cela, il inclut le Java Technology Deployment Toolkit, une suite d'outils et de fonctions permettant de simplifier la détection du JRE et de son installation. Surtout, avec le Consumer JRE, le JRE devient (enfin) modulaire. Modulaire, dans le sens, que le JRE installe uniquement ce que l'utilisateur a besoin pour faire fonctionner ses applications Java. C'est-à-dire que l'on télécharge (en transparence pour l'utilisateur) uniquement les modules dont l'application a besoin. Si par exemple, une nouvelle application a besoin de fonctions non instal-

lées dans le JRE, celui-ci va télécharger et installer les fonctions manquantes. Cela implique que le temps de démarrage sera rallongé la première fois, le temps d'installer. En procédant ainsi, on évite de télécharger un JRE de plus de 10 Mo ! Toujours sur le temps de démarrage, Consumer JRE doit fournir un processus de lancement plus rapide avec l'utilisation d'un cache disque en y chargeant des portions du JRE, ce qui doit réduire le temps de lancement d'une application.

La fin de l'enfer des classpath avec Java Module

Un des problèmes récurrents de Java concerne le classpath et le fameux "classpath hell". Cela intervient lorsque l'on a différents JRE / JDK installés, plusieurs versions de la même librairie, des mêmes classes, différents IDE Java, etc. Bref, comment gérer les versions ? Les développeurs Windows étaient habitués à gérer tant bien que mal le fameux "hell dll". Dans le monde Java, on parle souvent de *classpath hell*, de *jar hell* et de *Extension hell*. Pour remettre un peu d'ordre dans le déploiement et la gestion interne, on disposera de Java Module, qui

introduit le concept du "super jar". Il doit à la fois simplifier le développement et le processus de déploiement. Pour reprendre la formulation Sun (JSR 277), il s'agit de "copier" les principes de fonctionnement de Maven.

À quoi va servir Java Module ? L'ambition est de faire du versioning, c'est-à-dire de gérer les versions des classes, librairies, des JAR, et d'éviter des conflits entre différentes versions, installées dans différents répertoires. Il intègre une gestion de dépendance (de quelle classe et librairie ai-je besoin pour mon application), le tout reposant sur un référentiel local (avec sa base de données). C'est donc une avancée non négligeable pour les développeurs Java. Car aujourd'hui, le JAR ne gère pas le versioning.

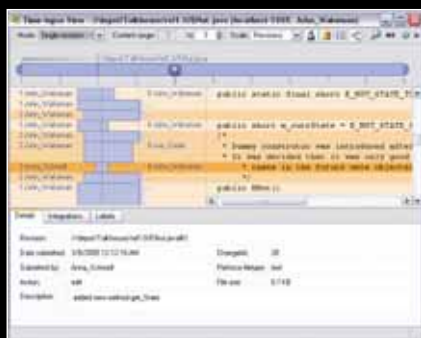
Pour ce faire, Java Module introduit les éléments suivants :

- JAM : nouveau format de Java Module, basé sur le format Jar. Il inclut en plus du Jar normal la signature du fichier jam (pour identifier, tracer le fichier), la compression des fichiers jam via Pack200, qualifiée d'hyper compression (JSR 200). Java 6 n'est que la première étape, Java 7 apportera de nouvelles fonctions



Présentation de la fonctionnalité Time-lapse View,

un atout de productivité du système de GCL Perforce.



Fonctionnalité Time-lapse View du logiciel Perforce

La fenêtre d'évolution temporelle (Time-lapse View) permet aux développeurs de voir chaque modification apportée à un fichier. Cette fenêtre offre une présentation dynamique sous forme d'annotations. Les développeurs peuvent enfin obtenir des réponses à leurs questions, notamment : « Qui a écrit ce code et quand ? » et « Quelles sont les modifications apportées au contenu et qu'est-ce qui justifie ces changements ? ».

L'outil Time-lapse View se présente sous la forme d'une ligne de temps qui rend compte, de manière claire, de l'évolution d'un fichier, modification après modification. Les dégradés de couleur reflètent l'âge du contenu du fichier et la ligne de temps peut être configurée pour indiquer les modifications apportées au fichier par numéro de révision, par date ou par numéro de modifications.

La fonctionnalité Time-lapse View n'est qu'un des nombreux atouts de productivité offerts par le système de GCL Perforce.

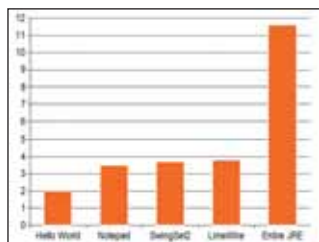


Jam, notamment sur la sécurité.

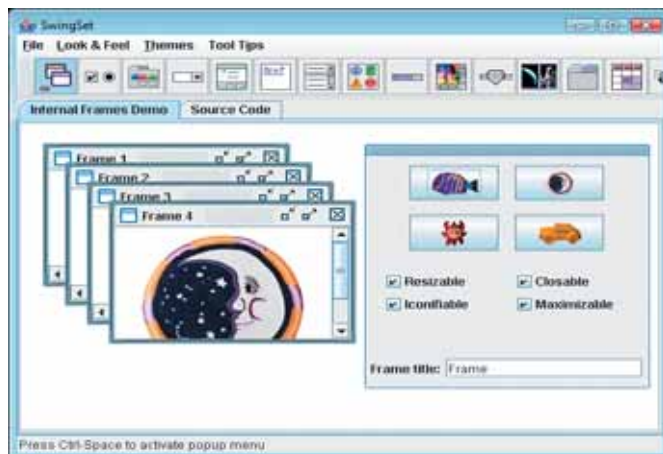
- Repositories : le référentiel est un " Maven like ". On dispose par exemple d'une hiérarchie.
- Support runtime : nouveau mécanisme durant le démarrage des applications, toutes les fonctions seront exécutées par la machine virtuelle et non dans un code tiers. Le runtime aura par exemple la validation des modules permettant de mieux contrôler les ressources et les modules réellement utilisés par l'application.

Sur Java Plug-In, l'implémentation a été entièrement revue. Le développeur pourra mieux spécifier un JRE spécifique pour exécuter une applet et les applets signées sont supportées dans une exécution Vista. Il est compatible avec IE 6 et 7 (XP et Vista). Il supporte Firefox 3 mais pas Firefox 2 et aucun support officiel n'est prévu selon les documents officiels. L'ancien Java Plug-in (ou Classic Java Plug-in) est toujours présent et on peut basculer d'un modèle à un autre.

Java Kernel : au cœur du futur de Java



Comme vu ci-dessus, le Consumer JRE a pour but d'être modulaire et de permettre un déploiement adapté du JRE. Le cœur de cette idée s'appelle Java Kernel. L'idée, vous l'aurez compris, est de proposer un JRE minimum avec le code et les fonctions pour fonctionner et nécessaires pour faire fonctionner la JVM et les applications. Jusqu'à présent, le JRE se composait de centaines de bundles se téléchargeant un par un. Le but est donc de proposer un " core java " rassemblant ces bundles de fonctionnement et qui soit le plus petit possible,



Les nouveautés Swing

entre 2 et 3 Mo. Et au fur et à mesure des besoins, le JRE téléchargera en arrière-plan les bundles nécessaires.

Du nouveau chez Swing

La partie graphique, souvent présentée comme une faiblesse, connaît une mise à jour avec Java 6 Update N. Cela aura un impact conséquent quant à l'exécution Java sur Windows avec une réécriture du pipeline graphique afin de prendre en compte Direct3D, ce qui permettra d'améliorer les rendus, et de mieux supporter la transparence. Même Java 2D pourra profiter de ce support et de l'accélération graphique matérielle. Concernant l'apparence, on bénéficie d'un nouveau Look and Feel nommé Nimbus, qui est une mise à jour des thèmes Ocean et Metal. Cette mise à jour permet de mettre à niveau les thèmes avec l'apparence des systèmes Vista et MacOS X 10.5 (et les nouveaux Linux).

Java 7 : 2009

À l'heure où nous écrivons, le JSR lié à Java 7 n'était pas encore créé, signifiant que la liste des

fonctions propres à Java 7 demeure ouverte et susceptible de bouger. Cependant, comme nous l'a précisé Alexis Moussine-Pouchkine (Sun France), l'expérience tirée de Java 5 permet de savoir ce qu'il faut faire et ne pas faire quand on modifie en profondeur l'architecture et les fonctions du langage. " Si les annotations ont été un succès, ce n'a pas été le cas des " generics " indique Alexis. Car si les développeurs utilisent volontiers les generics, les écrire reste souvent une corvée. Il existe d'ailleurs un débat autour des modifications langage à faire, notamment sur les " closure ". Une des " craintes " étant d'apparaître comme suiveur de C#... Java 7 est comparable, en termes de nouveautés, de modifications, à Java 5. Par comparaison, Java 6 se veut dans la continuité de Java 5, même si la Upgrade N comporte des rajouts importants (mais pas au niveau langage). Et si Java 6 reste un projet propriétaire, Java 7 sera la première vraie version open source (via le projet OpenJDK) de l'histoire de Java ! Côté agenda, Java 7 sera disponible courant 2009,

SE 6 == VICTORY!



peut-être durant le 1er semestre, soit un peu plus de 2 ans après la sortie de Java 5. Si Java 5 a été bien adoptée, malgré une lente percée à cause du manque de support dans les outils de développement, Java 6 l'a été par environ 20 % des développeurs (dixit Alexis). Sun s'attend d'ailleurs à des migrations directes Java 5 vers Java 7, qui reprendra les bases de Consumer JRE pour les étendre. De nombreuses nouveautés et améliorations sont attendues dans Swing (Swing Application Framework notamment), évolution des génériques, annotations étendues, disponibilité de NIO 2 (JSR 203), amélioration du support XML, support du javabeans property, des closures, de JMX 2, etc.

Java EE 6 : courant 2009

Sur le prochain Java EE, la v6, les choses évoluent elles aussi. Sur la v5, cela se déroule plutôt bien, même si on attend l'implémentation JBoss et Websphere (hors version community). Java EE 6 apporte les nouvelles versions suivantes : Servlet 3.0, JPA 2 (désormais disponible séparément des EJB), EJB 3.1, amélioration autour de JCA. La disponibilité est attendue pour le 1er trimestre 2009. Et l'implémentation de référence sera disponible en même temps (Glassfish v3).

JavaFX : trop immature

Présenté en fanfare à la JavaOne 2007, JavaFX était présenté comme une alternative aux plates-formes Adobe et Microsoft. Depuis, la technologie souffre de son immaturité et on en parle très peu. De l'aveu même de Sun, il n'y a pas grand-chose de nouveau depuis le printemps dernier. Mais les développeurs travaillent beaucoup à améliorer la plate-forme, le langage de script. La disponibilité prochaine de Consumer JRE aidera sans aucun doute JavaFX.

■ François Tonic

Extrême
Java

Seam

UML

Hibernate

valtech
training

Gestion
de projet

Scrum

XML

.Net

Au plus court vers vos nouvelles compétences

Architecture et intégration

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

Développement Java et C++

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

Microsoft .Net

- C# et la conception objet (5 jours)
- Programmation avec Visual Basic .Net et conception objet (5 jours)
- Programmation intensive avec le Framework .Net (5 jours)
- Développement d'applications Web avec ASP.NET (5 jours)
- Développement d'applications Windows Forms sur la plate-forme .Net (5 jours)
- Développer des applications Web Services avec .Net (4 jours)
- Développer des applications avec C# et le Framework .Net 3.0 (5 jours)

Gestion de projet

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

Frameworks Java EE

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

Analyse, conception et modélisation avec UML

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

Oracle

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

XML et Web Services

- Introduction à la technologie XML (1 jour)
- Introduction aux technologies Web Services (1 jour)
- Développer avec XML (3 jours)
- Développer une application XSL (2 jours)
- Développer des applications Web Services en Java (4 jours)
- Développer des applications XML avec Java (2 jours)

90 formations au développement logiciel

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

Développement Web

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

Stratégies de développement logiciel

- Test Driven Requirement ou la gestion des exigences dirigée par les tests (2 jours)
- Test Driven Development ou la programmation pilotée par les tests en Java (3 jours)
- Stratégie de test, vérification et validation (3 jours)
- Les fondements de l'IT Infrastructure Library (ITIL) (3 jours)
- Introduction au CMMI (3 jours)
- Gestion de configuration avec Rational ClearCase UCM (Unified Change Management) (2 jours)
- Gestion de configuration avec Rational ClearCase (version Windows ou Unix) (2 jours)
- Administration de Rational ClearCase (version Windows et Unix) (2 jours)
- L'usine logicielle, des concepts à la pratique (3 jours)

PHING : du ANT pour votre code PHP

Phing est un outil de maintenance basé sur Ant et écrit en PHP. Le fonctionnement de Phing est similaire à celui de Ant ou make et ses fonctionnalités particulièrement adaptées aux opérations de construction, nettoyage, test, déploiement et sauvegarde d'applications PHP. Cet article vous propose de découvrir l'intérêt et le fonctionnement de cet outil à travers quelques exemples simples.

Phing propose une architecture basée sur un outil Java : un méta-langage XML permettant de mettre en oeuvre des macro-commandes et un exécutable en ligne de commandes qui se manipule comme l'utilitaire Gnu make. Ces caractéristiques apportent de la rigueur qui facilite le travail à plusieurs, la pérennité des développements et l'homogénéisation des tâches de maintenance. En revanche, Phing nécessite l'apprentissage de son fonctionnement et du méta-langage associé, principale barrière à son adoption.

Cahier des charges

Nous nous proposons ici d'utiliser Phing dans le cadre de la maintenance d'une petite application PHP. Nous considérons que l'application en question est répartie sur trois serveurs : développement, pré-production et production. Les cibles que nous allons développer seront les suivantes :

- Aide de l'utilitaire (help)
- Nettoyage de l'application en développement (clean)
- Sauvegarde de l'application en développement (save)
- Déploiement en pré-production (install)
- Synchronisation de la pré-production à la production (sync)

Un peu de vocabulaire

Une fois Phing installé, la commande phing est disponible. L'exécution de celle-ci nécessite la présence d'un fichier contenant des directives XML qui décrivent les actions à réaliser (par défaut ce fichier s'appelle build.xml), tout comme make nécessite un fichier Makefile. Ce fichier contient un tag *racine project* qui englobe plusieurs éléments parmi lesquels :

- Les cibles, qui sont des points d'entrée sur les actions à réaliser.
- Les tâches, qui constituent des opérations élémentaires utilisées par les cibles.
- Les propriétés, qui permettent de paramétrer ou d'obtenir des informations.
- Les types et les fileset, qui se comportent comme des conteneurs d'information (liste de fichiers par exemple).
- Les filtres, chaînes de filtres (filterchains) et mappers qui seront utiles dans les manipulations des noms et des contenus de fichiers.

Il est également possible d'étendre Phing à l'aide de code PHP en proposant des tâches, des types et des mappers personnalisés.

Installation de Phing

Passons à la pratique ! Il est possible d'installer Phing manuellement ou à l'aide de PEAR. Cette dernière méthode est la plus immédiate. Une fois votre environnement PHP installé, il vous suffit de taper les lignes suivantes :

```
pear channel-discover pear.phing.info
pear install phing/phing
```

Sous Windows, vous trouverez l'exécutable pear.exe dans le répertoire d'installation de PHP. Si vous ne trouvez pas cette commande et qu'il

existe un script go-pear.bat, double-cliquez dessus pour installer l'environnement pear. Ces commandes ont pour effet d'installer les sources de phing et la commande phing qui sera utile à l'utilisation de l'outil. Vous pouvez modifier la variable \$PATH de votre environnement pour rendre cette commande disponible partout.

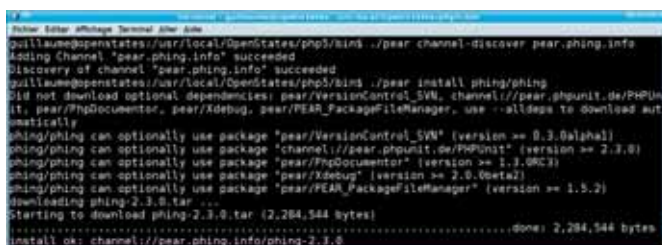


Figure 1 : Installation de Phing avec PEAR

1 La cible " help "

Nous allons nous placer à la racine de notre application PHP. Pour que cet exemple soit utile, vous pouvez l'appliquer à une de vos applications. Dans ce répertoire, nous allons créer un fichier *build.xml* contenant la balise *racine project* et une cible *help* qui affiche un message d'aide. Voici le contenu du fichier :

Le fichier build.xml

```
<?xml version="1.0"?>
<project name="OpenStates" default="help" basedir="." >
  <target name="help">
    <echo msg="Available targets : " />
    <echo msg=" - help ..... this message" />
    <echo msg=" - clean .... clean the environment" />
    <echo msg=" - save ..... backup current application" />
    <echo msg=" - install .. deploy stable svn tag in preprod" />
    <echo msg=" - sync ..... rsync preprod to prod" />
  </target>
</project>
```

help est notre cible par défaut (default="help") et toute cible s'appliquera par défaut au répertoire courant (basedir="."). L'exécution de la commande phing devrait afficher le message d'aide correspondant. Si c'est le cas, nous pouvons passer à l'étape 2 (Fig.2).

2 La cible " clean " (Fig.3)

Cette cible va nous permettre de supprimer les fichiers inutiles : fichiers temporaires, cache, etc. Pour cela, nous allons ajouter une cible dans *build.xml* juste avant la balise de fermeture *</project>* tel que le montre l'exemple ci-après.

La cible clean


```
...
<!-- clean environnement -->
<target name="clean">
```


**Le rendez-vous incontournable
des développeurs et professionnels de l'informatique !**

Microsoft **TechDays2008** *L'innovation avance avec vous*

Retrouvez toutes les innovations informatiques présentes et à venir réunies dans un événement unique et plus que jamais incontournable.

Édition Spéciale Lancement

 **Windows Server[®] 2008**

Microsoft[®]
SQL Server[™] 2008

 Microsoft[®]
Visual Studio[®] 2008

**11, 12, 13
février 2008**

**au Palais des
Congrès de Paris**

***Circuit recrutement
Microsoft et partenaires !***

À vous de sélectionner votre parcours :

22 parcours techniques

220 sessions sur tous les produits,
toutes les solutions Microsoft
et plus

... Datacenter | Sécurité | Développement web | Décisionnel | Windows Vista | SharePoint...

msdn  **Microsoft TechNet**

Inscrivez-vous, c'est gratuit !
www.microsoft.com/france/mstd08

```
guillaume@openstates: /web/apps/openstate
guillaume@openstates:/web/apps/openstates.com$ phing
Buildfile: /media/sda6/web/apps/openstates.com/build.xml

OpenStates > help:

[echo] Available targets :
[echo] - help ..... this message
[echo] - clean ..... clean the environment
[echo] - save ..... backup current application
[echo] - install .. deploy stable svn tag in preprod
[echo] - sync ..... rsync preprod to prod

BUILD FINISHED

Total time: 0.0898 seconds

guillaume@openstates:/web/apps/openstates.com$
```

Figure 2 : exécution de la commande phing

```
guillaume@openstates: /web/apps/openstates.com
guillaume@openstates:/web/apps/openstates.com$ phing clean
Buildfile: /media/sda6/web/apps/openstates.com/build.xml

OpenStates > clean:

[echo] Cleaning up...
[delete] Deleting 6 files from .
[delete] Deleting /media/sda6/web/apps/openstates.com/.www/bout-practique.php.jpg.bak
[delete] Deleting /media/sda6/web/apps/openstates.com/.blog/cache/26376.php
[delete] Deleting /media/sda6/web/apps/openstates.com/.blog/cache/26356.php
[delete] Deleting /media/sda6/web/apps/openstates.com/.blog/cache/26383.php
[delete] Deleting /media/sda6/web/apps/openstates.com/.blog/phpinfo.tmp
[delete] Deleting /media/sda6/web/apps/openstates.com/.build.xml.tmp

BUILD FINISHED

Total time: 0.3077 seconds

guillaume@openstates:/web/apps/openstates.com$
```

Figure 3 : exécution de la cible clean

```
<echo msg="Cleaning up..." />
<delete verbose="true">
  <fileset dir=".">
    <include name="**/*.tmp" />
    <include name="**/*.bak" />
    <include name="**/cache/**" />
    <exclude name="**/.htaccess" />
  </fileset>
</delete>
</target>
```

Nous supprimons ici l'ensemble des fichiers ayant des extensions *tmp* ou *bak* ainsi que ceux contenus dans les répertoires *cache* excepté *.htaccess*. Ces règles sont inscrites dans le *fileset* contenu dans la tâche *delete*. La présence d'un double *** signifie que l'on va chercher les fichiers récursivement dans la hiérarchie du dossier courant.

3 La cible "save" (Fig.4)

Cette fois nous allons introduire deux notions supplémentaires : les *fileset indépendants* et *propriétés*. L'avantage d'un *fileset* déclaré de manière indépendante, c'est-à-dire en dehors d'une tâche ou d'une cible, est de pouvoir être réutilisé. La déclaration du *fileset* se fera au début du fichier *build.xml* et celle de la cible *save* se fera à la fin, à la suite de la cible *clean*. *

Le fileset backupfiles et la cible save

```
<!-- fileset for files to backup -->
<fileset dir="." id="backupfiles">
  <include name="blog/**" />
  <include name="www/**" />
```

```
guillaume@openstates: /web/apps/openstates.com
guillaume@openstates:/web/apps/openstates.com$ phing save
Buildfile: /media/sda6/web/apps/openstates.com/build.xml

OpenStates > save:

[echo] Creating archive...
[tar] Building tar: /media/sda6/web/apps/openstates.com/backups/1137679182_OpenStates.tar.gz

BUILD FINISHED

Total time: 1.0104 seconds

guillaume@openstates:/web/apps/openstates.com$
```

Figure 4 : exécution de la cible save

```
</fileset>
...
<!-- backup -->
<target name="save">
  <echo msg="Creating archive..." />
  <tar destfile="/backups/${env.REQUEST_TIME}_${phing.project.name}.tar.gz"
    compression="gzip" basedir="." />
  <fileset refid="backupfiles" />
</tar>
</target>
...
```

Une propriété est un ensemble clé/valeur. Ici, nous utilisons deux propriétés natives de Phing : *\${env.REQUEST_TIME}* et *\${phing.project.name}* qui sont respectivement substituées par un timestamp UNIX et le nom du projet Phing. Cela permet de personnaliser le nom du fichier de sauvegarde.

4 La cible install

Cette cible a pour objectif d'installer en pré-production la version stable d'une application. Nous allons ici utiliser une tâche dite optionnelle, c'est-à-dire dépendante de la présence de l'outil que nous allons utiliser, en l'occurrence Subversion.

```
guillaume@openstates: /web/apps/openstate
guillaume@openstates:/web/apps/openstates.com$ phing install
Buildfile: /media/sda6/web/apps/openstates.com/build.xml

OpenStates > delpp:

[echo] Cleaning preprod...
[delete] Deleting 23 files from /mnt/preprod/www

OpenStates > install:

[echo] Installing...
[svnexport] Exporting SVN repository to '/mnt/preprod/www/'

BUILD FINISHED

Total time: 0.1672 seconds

guillaume@openstates:/web/apps/openstates.com$
```

Figure 5 : exécution de la cible install

Une cible supplémentaire *delpp* nous permettra de supprimer les fichiers de l'ancienne version de l'application en pré-production. La cible *install* sera liée à la cible *delpp* par une relation de dépendance (argument *depend="delpp"* de la cible *install*). Cela aura pour effet d'exécuter la cible *delpp* automatiquement avant l'appel de *install*. La cible *install* va exporter la version stable de l'application extraite d'un dépôt de données Subversion. Nous considérons ici que le serveur de pré-production est accessible depuis le serveur de développement (partage réseau). **La cible install**

```
<!-- preprod remove -->
```


LEONARDI

ihm

EN TOUTE SIMPLICITÉ !

ATELIER
"QUICK START"
LEONARDI
21 ET 22 FEV.
500€ HT
REMBOURSABLES.

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

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

NOUVELLE
VERSION
LEONARDI
V4.0
OPEN-SOURCE

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



LYRIA

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



```
<target name="delpp">
  <echo msg="Cleaning preprod..." />
  <delete>
    <fileset dir="/mnt/preprod/www">
      <include name="*" />
    </fileset>
  </delete>
</target>

<!-- preprod deploy -->
<target name="install" depends="delpp">
  <echo msg="Installing..." />
  <svnexport>
    svnpath="/usr/bin/svn"
    force="true"
    nocache="true"
    repositoryurl="file:///usr/local/svn/tag/STABLE"
    todir="/mnt/preprod/www/" />
  </target>
...
```

Cette opération est dépendante du paquetage PEAR VersionControl_SVN. Si vous ne l'avez pas, Phing va échouer et vous demander de l'installer.

5 La cible sync

Pour cette étape, nous devons effectuer une synchronisation avec l'outil *rsync* qui n'est malheureusement pas proposé dans les tâches optionnelles de phing. La tâche *exec* nous permettra d'effectuer un appel à la commande *rsync* du système d'exploitation.

La synchronisation à effectuer se fera depuis le serveur de pré-production. Sur le serveur de production, un démon *rsync* doit être installé et paramétré avec un fichier */etc/rsyncd.conf*, comme le montre l'exemple ci-après. Si vous n'avez qu'une machine, elle fera office de pré-production et de production.

Le fichier */etc/rsyncd.conf* (serveur de production)

```
uid = rsync
gid = rsync
use chroot = no
max connections = 2
syslog facility = local5
pid file = /var/run/rsyncd.pid
log file = /var/log/rsyncd.log

[openstates]
  path = /web/openstates
  comment = OpenStates en production
  write only = true
  read only = false
  list = false
  uid = guillaume
  gid = guillaume
  exclude *.bak *.tmp *-
  secrets file /etc/rsyncd.secrets
  hosts allow preprod
```

Le fichier */etc/rsyncd.secrets* ne doit être lisible que par le démon *rsync* (chown *rsync* */etc/rsyncd.secrets* ; chmod 400 */etc/rsyncd.secrets*).

Une fois l'outil installé, il reste à remplir le fichier *build.xml* avec la commande *rsync* adéquate. Pour cela, nous allons créer une cible *sync* qui contiendra une tâche *exec* :

La cible sync

```
...
<!-- synchronize preprod->prod -->
<target name="sync">
  <echo msg="Synchronize..." />
  <exec command="rsync -a -password-file=/etc/rsync.pass
    -delete -delete-after -delete-excluded
    /mnt/preprod/www/ preprod::openstates/
    2> /var/log/syncerr.log"
    escape="false" />
</target>
...
```

Le fichier */etc/rsync.pass* (serveur de pré-production)

```
mon_password
```

La cible *sync* effectue une synchronisation du répertoire */mnt/preprod/www* en pré-production vers le répertoire */web/openstates* en production. Il s'agit d'une synchronisation de type miroir, c'est-à-dire que toute modification, ajout ou suppression en pré-production sera répercutée en production lors de la synchronisation.



Figure 6 : exécution de la cible sync

Pour aller plus loin

Cet article est une introduction au fonctionnement de Phing. Il existe encore plein d'autres possibilités liées aux manipulations de fichiers et de leur contenu ou encore aux tâches optionnelles *PHPUnit*, *PEAR*, *IonCube* ou *ZendAnalyser*. La documentation en ligne propose pour chaque fonctionnalité une description et un exemple.

- <http://phing.info/docs/guide/current/>

Comme nous avons pu le découvrir ici, Phing propose non seulement des fonctionnalités permettant de paramétrer la maintenance d'un environnement d'exécution PHP mais aussi un cadre de travail précis qui peut être réutilisé d'un projet à l'autre. Vous pouvez télécharger les fichiers de cet article et poser vos questions pour aller plus loin à l'adresse suivante : <http://blog.openstates.com/phing/>.

■ Guillaume Ponçon

Auteur de l'ouvrage *Best practices PHP5*, ingénieur et consultant PHP chez OpenStates en partenariat avec Zend Technologies France, Guillaume est spécialisé dans l'industrialisation et l'optimisation d'applications PHP professionnelles.

guillaume.poncon@openstates.com

<http://www.openstates.com>


```
saveArticle delegates to ArticleRepository.save;
// Délègue au Repository les opérations de recherche
findAuteurByName delegates to
    AuteurRepository.findAuteurByName;
List<@Article> findArticleByTitre(String baseArticles,
String articleName);
findAuteurByName delegates to
    AuteurService.findAuteurByName;
}

Repository AuteurRepository {
    int getNumberOfAuteurs() delegates to GenericDAO;
    save;
    findByQuery;
    List<@Media> findAuteurByName(String name);
}
```

On remarque la présence des trois briques essentielles du DDD : Entité, Repository et Services.

Les services s'appuient sur les méthodes des entités qui s'appuient elles-mêmes sur leur Repository pour résoudre les dépendances. Concrètement, lorsqu'un auteur a besoin de calculer son taux de productivité, il appelle la méthode *calculeProductiviteIAuteur()* présente dans l'entité Auteur qui appelle la méthode *findArticleByAuteur()* à partir du Repository Auteur. La rupture avec l'approche orientée services classique intervient dès lors qu'il existe un couplage entre la méthode *findArticleByAuteur()*, le Repository et l'entité.

On s'aperçoit alors que le DDD centralise l'ensemble du métier, mais aussi indirectement, l'ensemble des objets d'accès aux données dans la couche du domaine.

D'un point de vue technique, on imagine aisément qu'en fonction de la plate-forme cible, il est possible de choisir la meilleure technologie. Par exemple :

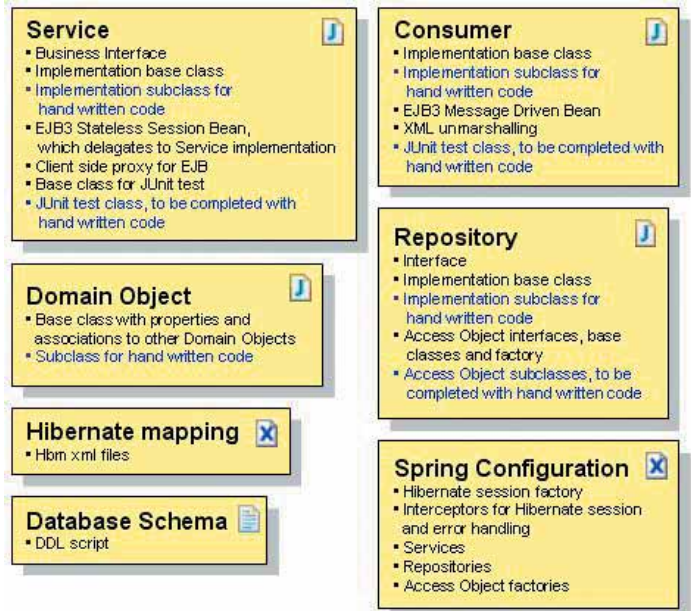
- Pour .NET, les choses sont un peu plus compliquées du fait de la présence de WCF comme serveur d'application. WCF est un ESB fonctionnant sous la forme d'échange de messages. Le DDD et son approche Rich Domain Object nécessite un serveur d'application sachant véhiculer des objets (les entités) entre couches. Une solution serait *Linq for Entities* comme Factory, WCF pour l'hébergement de la couche de services, et les contrats de données (DataContract) pour représenter les entités.
- Pour JEE : La couche de services sera représentée par des EJB Session de type POJO (Plain Old Java Bean). La couche Repository s'appuiera sur des usines de type Entity Manager avec JPA (Java Persistence API) pour rechercher les entités (dans un outil de mapping tel que Hibernate, Toplink ou Kodo). Les entités seront des EJB Entités traditionnels.

Les outils

Le DDD est une pratique encore naissante. Très peu (voire pas du tout) utilisée en France, elle est représentée d'une certaine manière par l'immense majorité des développeurs qui s'appuient sur les pratiques de développement multi couche.

Ceux qui s'en rapprochent le plus sont ceux qui ont essayé de générer des architectures n-tiers à partir de modèles. En .NET, on peut citer

dans cette catégorie le projet Candle développé par Alain Metge (<http://www.dsifactory.org/photos/alain/images/115/original.aspx>) ou en plus industrialisé, le projet Sculpitor Java.



Les concepts DDD dans sculpitor.

Partir d'un DSL pour générer une architecture DDD n'est bien entendu pas un passage obligé. Il existe de nombreux exemples de codes sur lesquels il est possible de s'appuyer pour concevoir et développer directement à la mode DDD. Le site <http://colinjack.blogspot.com/> est un exemple du genre pour .NET. Non content de proposer une application de e-commerce en référence (téléchargeable sur le site), l'auteur y traite au quotidien des problématiques du DDD. A noter que la communauté .NET fait partie des plus actives en s'appuyant sur le duo Castle + NHibernate et en proposant de nombreux patterns de conception.

Le DDD objet de nombreux débats

Sous les feux des projecteurs et aidé en cela par une communauté très active, le DDD suscite de nombreuses interrogations et débats, parfois enflammés, toujours passionnés. Le principal reproche qui lui est fait tient à la place, jugée trop importante, laissée à la couche du domaine. Certains verraient bien un juste milieu entre une couche anémique et une couche riche. Mais Eric Evans y tient, ce qui fait la force du DDD est le spectre (très large) des responsabilités données aux entités, allant du dialogue avec les Repositories à l'implémentation des règles de gestion.

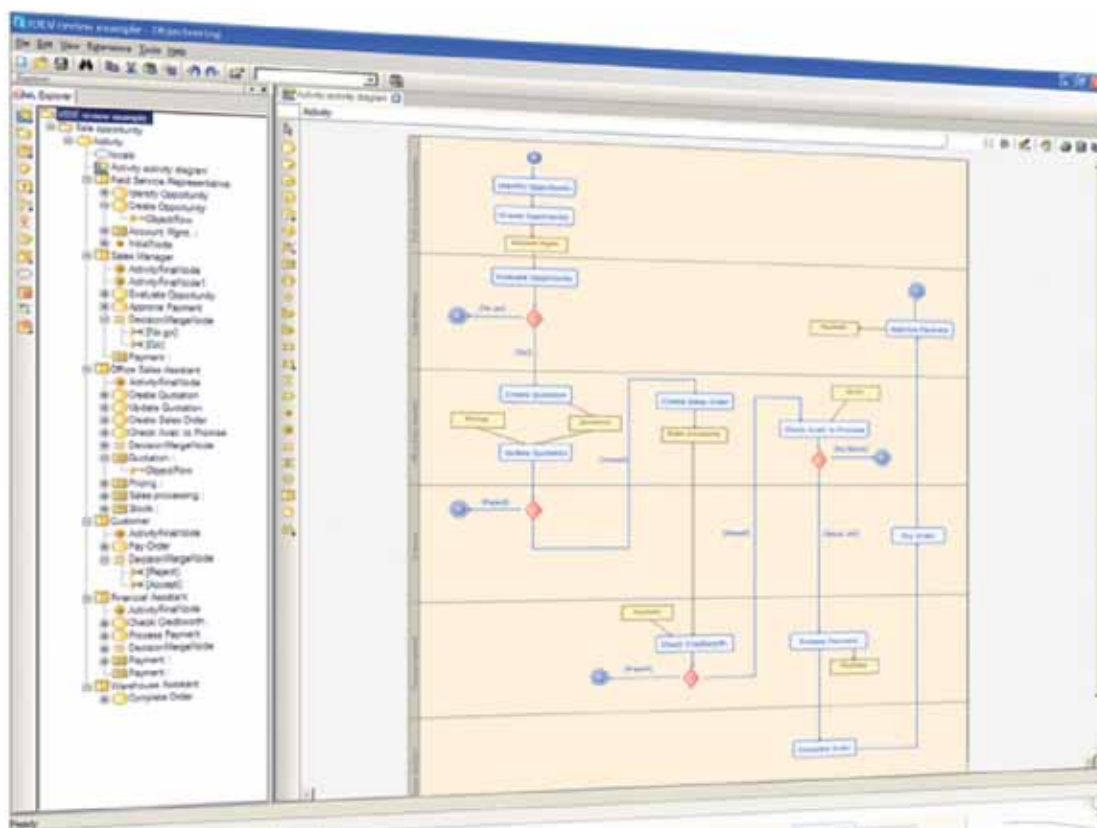
Pour autant, rien n'est encore joué. Si le DDD se verrait bien représenter l'architecture d'entreprise idéale du futur, il lui faudra gagner les voix des nombreux sceptiques, dont les créateurs d'outils ORM eux-mêmes. Tel un Active Record et sa trop forte proximité avec la couche d'accès aux données, Le DDD est critiqué. Les premières références réussies lui donneront peut-être une légitimité.

■ Sami Jaber

Webmaster du site DotNetGuru.org et fondateur de la société DNG Consulting

Your projects deserve a tool*

Objecteering Modeler



Objecteering 6.1

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

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

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

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

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

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

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

Architecture
d'Entreprise

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

Objecteering
SOFTWARE

The model-driven development company

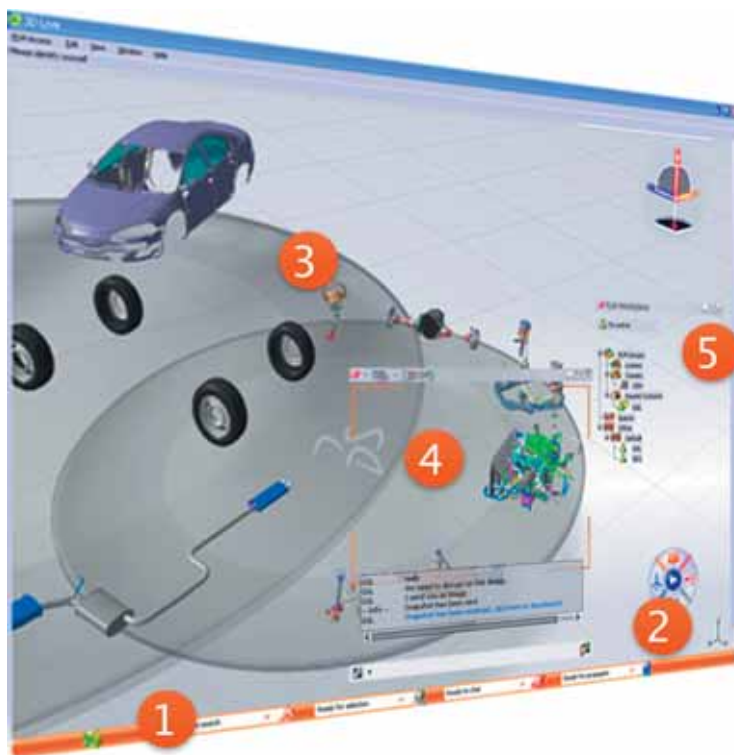
* Vos projets méritent un outil

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

Microsoft 2008

Tout pour le développeur !

Depuis novembre 2007, les développeurs sont particulièrement gâtés par Microsoft : framework .Net 3.5, Visual Studio 2008, gamme Express 2008. Dans quelques semaines, ce sera au tour de Windows Server 2008 et d'ici juin prochain : les nouvelles API Live et surtout SQL Server 2008 ! Si cela ne suffisait pas, on nous rajoute : Silverlight 2, PowerShell 2, Astoria, Expression 2, Team System 2008, les extensions ASP.Net, un nouveau Popfly, etc., l'année 2008 sera très très riche. Mettez-vous dès maintenant au millésime 2008 !



ADO.Net Data Services (alias Astoria) et les architectures REST

Très discret mais terriblement important dans les nouvelles architectures applicatives, Astoria est, pour nous, un nouvel élément central dans les approches services, SOA, applications Ajax, etc. Initié au sein de l'équipe SQL Server, Astoria est un framework d'exposition d'un modèle de données via URI, tout en reposant sur les concepts REST. Car finalement, URI, GET, Post, Put et Delete, suffisent à l'écrasante majorité des applications pour requêter, accéder, exposer les données. Astoria expose les données en XML ou JSON (via un processus de transformation). Le mapping se fait via une librairie cliente qui réalise un map des données sous forme d'objets .Net, ce qui simplifie bien les choses pour le développeur et facilitera sans aucun doute le travail dans des applications riches (WPF, Winforms, Silverlight 2, etc.). Et cerise sur le gâteau, Astoria s'interface avec tout objet implémen-

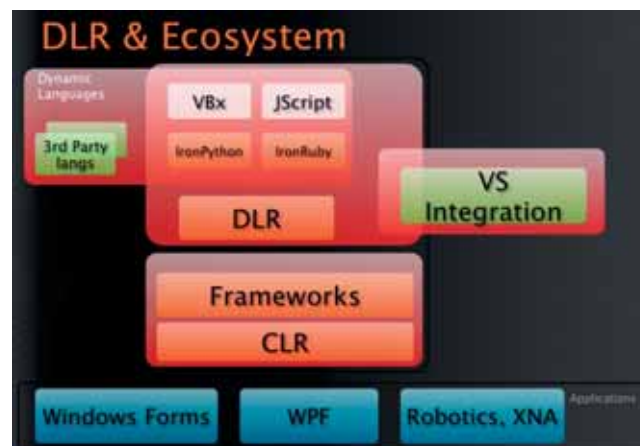
Depuis quelques numéros de Programmez, ainsi que du Hors Série .Net, nous vous dévoilons au fur et à mesure les nouveautés Microsoft pour le développeur. Nous avons déjà assez largement évoqué .Net 3.5, Linq et Visual Studio. Dans notre dossier spécial, nous continuons sur notre lancée avec Visual Studio Shell ou comment faire sa propre distribution à la Eclipse, PowerShell 2.0, le framework Sync, la gamme Expression, les bonnes pratiques de migration ou encore Volta, un projet qui fait beaucoup parler de lui !

En réalité, Microsoft a quasiment établi à la TechEd les premières lignes directrices jusqu'en 2009-2010 en dévoilant ici et là l'agenda de plusieurs langages et outils. Cet agenda sera largement complété lors de la prochaine PDC (PDC '08) à l'automne prochain, qui fournira une vision globale du (possible) futur des technologies selon Microsoft.

Toujours plus de cohérence

Avec les dernières versions des outils et logiciels de l'éditeur, .Net termine lentement mais

sûrement d'être la plate-forme technique commune. Il fait une apparition discrète dans Windows Server 2008, mais c'est surtout le cas dans Biztalk qui intègre, enfin, .Net. Office 2007 a fait de même en s'appuyant assez largement dessus et en renouvelant entièrement la conception applicative directement dans Visual Studio. On constate le même mouvement dans les gammes Dynamics. Par ailleurs, au-delà des outils et du cœur .net, l'éditeur complète sa plate-forme par des extensions, des frameworks. C'est aussi la force de .net : la modularité et l'extensibilité. Et en 2008, nous aurons notre lot de nouveautés. En premier lieu, le framework Sync, que nous abordons longuement dans ce dossier. Stratégique, Sync est l'offre de synchronisation des données pour les applications. Surtout, il est capable de gérer le mode connecté et déconnecté, fonction absente aujourd'hui si ce n'est avec de grands artifices.



tant lqueryable de Linq ! Astoria a donc pour ambition, en environnement .Net, de fournir une architecture claire et un minimum unifiée, car aujourd'hui, chaque application web à son modèle de données : Ajax, Silverlight, les services ou encore les mashup. Comme vu plus haut, ce framework fournit une interface de données REST avec : un modèle Data Entity, URI, http, différents formats et différentes sources (Entity framework et Linq). La solution idéale dans une application web est de pouvoir se mapper directement à une source de données sans passer par un tiers, voilà l'intérêt d'Astoria. La disponibilité de la v1 est prévue vers mi-2008.

Du côté de Live

Windows Live reste encore marginal mais dans les mois à venir, le développeur aura à sa disposition de nouvelles versions d'API que l'on promet stables. Dès aujourd'hui, les services offerts dans Live sont nombreux : hot-mail, Live ID, Messenger, Search, Spaces, gadgets, earth, calendar, etc. On peut attaquer les services via des API, des contrôles pour les intégrer dans ses propres applications. Ainsi, si on prend Windows Live Spaces, on dispose des contrôles javascript (Spaces Photo), API d'accès WebDev, etc. En novembre dernier, Live s'est enrichi d'API pour Messenger IM, de la gestion de la présence. Là encore on reste sur des standards : javascript, REST, JSON, HTML... Sur la partie authentification (Live ID), on dispose d'un SDK à part : Windows Live ID Web Authentification SDK et de son client (basé sur des classes .Net 2). Virtual Earth connaît lui aussi un lifting pour le développeur avec contrôle javascript, API SOAP, web services. Tout comme pour Google, n'oubliez pas de vérifier les licences d'utilisation car dans

certains cas, une utilisation commerciale ou selon un volume donné, il faut passer à la caisse. Petite note française, n'oubliez pas de regarder, le projet bluePortal (disponible sur CodePlex). Il s'agit d'un partage vidéo et d'une plate-forme collaborative. Il se base sur un Core System (video metadata, encoding, uploading, etc.), starterkit (exemple d'un youtube like) et API web service. Il peut se déployer seul sur un site ou comme service d'un site existant.

Office 2007 : un nouveau départ pour VSTO et même VBA

Le développement dans Office 2007 n'a pas été oublié, au contraire. Avec le passage à Visual Studio 2008, l'outil VSTO a été refondu pour mieux intégrer ce développement dans l'IDE. Notons qu'avec VSTO 3, VBA a connu quelques modifications. Ainsi, on peut appeler du VBA dans VSTO et du code pur VSTO dans VBA. Mais ce n'est pas pour autant que la migration VBA ne soit pas conseillée... Les nouveautés VBA concernent surtout le support de .NET 3.5 et l'accès à des fonctions précises d'Office 2007 (essentiellement le ruban). Cependant, de nombreuses fonctions disponibles en VSTO ne le sont pas en VBA (ListObject, cache de données, contrôles managés dans un document, smart tags). Malgré tout, les appels de code facilitent la vie du développeur. Selon Microsoft, il faut cependant être très strict dans les noms utilisés. Dans le cadre d'un réemploi de code VBA, il est possible de le faire dans VSTO sans réécrire le code VBA existant, par contre, dans le cas de sous-routines à migrer dans .Net, il faudra réécrire. D'autre part, l'éditeur NE CONSEILLE PAS de migrer les projets VBA entièrement, excepté si le besoin se fait pressant.

2008 : année de la communication unifiée même pour le développeur !

2007 on en parlait. 2008 il faudra s'y mettre ! Si la communication unifiée impacte l'infrastructure, l'utilisateur, le serveur, le développeur ne sont pas en reste. Il s'agit de regrouper la conférence, la messagerie instantanée, la VoIP, la messagerie, la collaboration, le partage dans un seul et même ensemble doté d'une interface commune et de fondations communes, on casse les silos de la communication précédente. Pour le développeur, cela ne simplifie pas forcément les choses. On dispose pour chaque ensemble de communication, d'API, de contrôles, de SDK, en tout une dizaine. Il faut implémenter les API clientes, gérer les fonctions asynchrones (javascript, Ajax), implémenter les services vocaux, gérer la présence, implémenter SIP (via son langage), etc. Mais les capacités d'intégration dans ces applications de la communication unifiée peuvent offrir de nouveaux services et simplifier l'échange d'informations, de données, par exemple dans un espace collaboratif interactif, dans un contexte de collaborateur nomade, etc.

La mobilité au rendez-vous

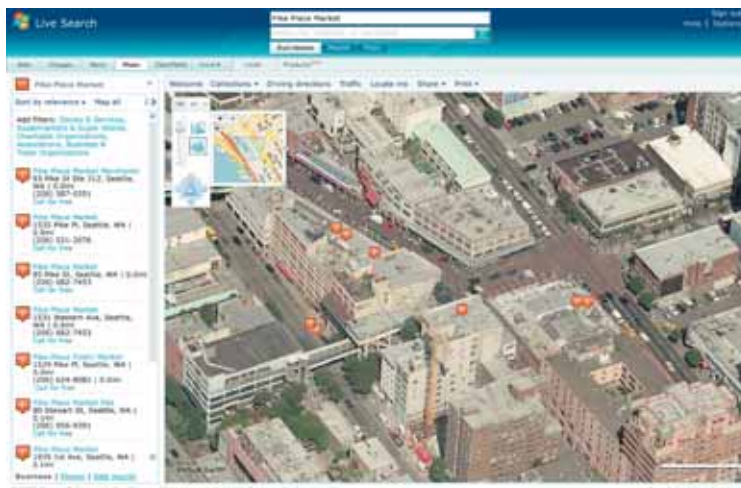
Le développement mobile est bien entendu présent dans la gamme 2008 et notamment avec le Compact Framework 3.5. Cette version se veut plus productive pour le développeur, un framework le plus réduit possible (env. 6,2 Mo), maintien des performances, compatibilité des anciens binaires (98 % selon l'éditeur) et avec le framework .net " normal ". Aujourd'hui, on dispose de Windows Communication Foundation, de Linq et du profiler CLR. Côté VS 2008, l'émulateur supporte l'ensemble des versions

mobiles (Compact, SQL Server, Windows Mobile et ce, dans les versions récentes et anciennes). Dans VS 2008, on bénéficie de meilleurs tests unitaires, d'un gestionnaire de sécurité, du SDK Mobile 5, SQL Server Compact Edition 3.5, de nouveaux compilateurs et d'un framework Core-Con managé. Côté Compact 3.5, on rajoutera aussi les nouveautés suivantes : support de la compression, API audio, Winforms améliorés, bug fix. Le côté application distribution est bien présent avec notamment la possibilité de faire du Machine to Machine. Sur le support de Linq : Linq to SQL, Linq to Entities et Expression trees sont indisponibles.

Des projets et évolutions à surveiller de très près !

Au-delà des outils et frameworks, 2008 sera marqué par la disponibilité de nombreux éléments. Citons les plus importants :

- **DLR (Dynamic Language Runtime)** : jusqu'à présent .Net est surtout du langage statique comme C# ou VB. Avec l'annonce de Silverlight 2.0 (anciennement 1.1) et de son implémentation .Net, on a vu apparaître la DLR. Elle permet d'utiliser directement dans des applications .Net des langages dynamiques, par exemple Ruby, comme le propose le monde Java avec Jruby. Il s'agit là d'une avancée importante, notamment pour les applications web. La version Linux sera disponible dans le projet Moonlight de Mono. La DLR sera disponible avec Silverlight 2.0. À noter qu'un VB dynamique est prévu.
- **ParallelFX** : nous l'avions évoqué dans le dossier programmation multicoeur. Il s'agit d'extensions à .Net pour implémenter la parallélisation dans son code (et dans Linq). Avec un minimum de modifications, on pourra paralléliser un code pour exploiter plusieurs cœurs. En cours de développement. Disponibilité de la v1 non précisée.
- **Silverlight 2.0** : comme indiqué précédemment, cette version va intégrer .Net, ce qui va sans doute aider Silverlight auprès des développeurs. Sur la version mobile toujours aucune nouvelle précise (peut être au prochain Mix du printemps).



- **Expression Studio 2.0** : cette version doit supporter WPF 3.5 (et VS 2008) et les applications Silverlight 1.0. Sur le support de Silverlight 2.0, pour le moment rien n'est prévu. La v2 notamment de Blend va grandement améliorer l'environnement : nouveaux objets 3D, nouvelles animations, options de build, synchronisation des fichiers échangés, etc. Disponibilité : 2e trimestre (?).
- **ASP.Net** : le côté web n'est pas oublié. ASP.Net connaît de profondes mutations. Cela passera notamment par l'apparition de ASP.Net MVC Framework qui permettra (enfin ! diront les développeurs) d'utiliser dans le langage le modèle MVC.

vNext : au-delà de 2008

Si pour cette année, on a déjà de quoi faire, ce n'est pas pour autant que les laboratoires de Microsoft ne planchent pas sur les prochaines versions et évolutions. Voici quelques nouveautés à prévoir :

.Net 4.0 (2009 ou 2010) : cette version devrait bénéficier de l'intégration de nouveaux frameworks. On peut s'attendre à l'apparition du projet Acropolis sur les applications composites (des éditeurs comme BEA, SoftwareAG et plus largement dans le monde Java le proposent ou le proposeront dans les mois à venir). Verra-t-on apparaître un C# 4.0, un nouveau VB ? L'introduction de F# pour la programmation fonctionnelle ? Disponibilité : aucune date n'a été annoncée. Il est possible que .Net 4.0 soit intégré en standard dans le futur Windows, 7 (anciennement Vienna), dont la disponibilité est prudemment annoncée pour fin 2009 ou 2010. Sur la compatibilité avec l'existant et les nouveautés, il faudra patienter quelques mois... Des discussions se font jour sur la

possibilité d'intégrer directement dans le compilateur l'obfuscation de code. Possibilité d'intégrer directement dans les classes des périphériques (comme des caméras...). Le multicore et la programmation parallèle devraient faire une entrée retentissante, mais sous quelle forme : intégration du Parallel FX, un nouveau compilateur dédié (voir le projet Phoenix) ou alors l'apparition d'un P# (P pour Parallel) ? Dans les pistes plus réelles : facilité d'utiliser IPv6, nouvelles méthodes de cryptage,

apparition d'un smart compiler, prévenir l'injection de dll.

VB NEXt : les premières tendances et orientations sont déjà connues, mais pas définitives. On devrait avoir l'intégration du dynamique dans le langage (pour le DLR), d'une simplification syntaxique, d'un plus haut niveau d'abstraction. Une réflexion de fond existe aussi sur les notions de programmation concurrente, sur le modèle de données, les workflows.

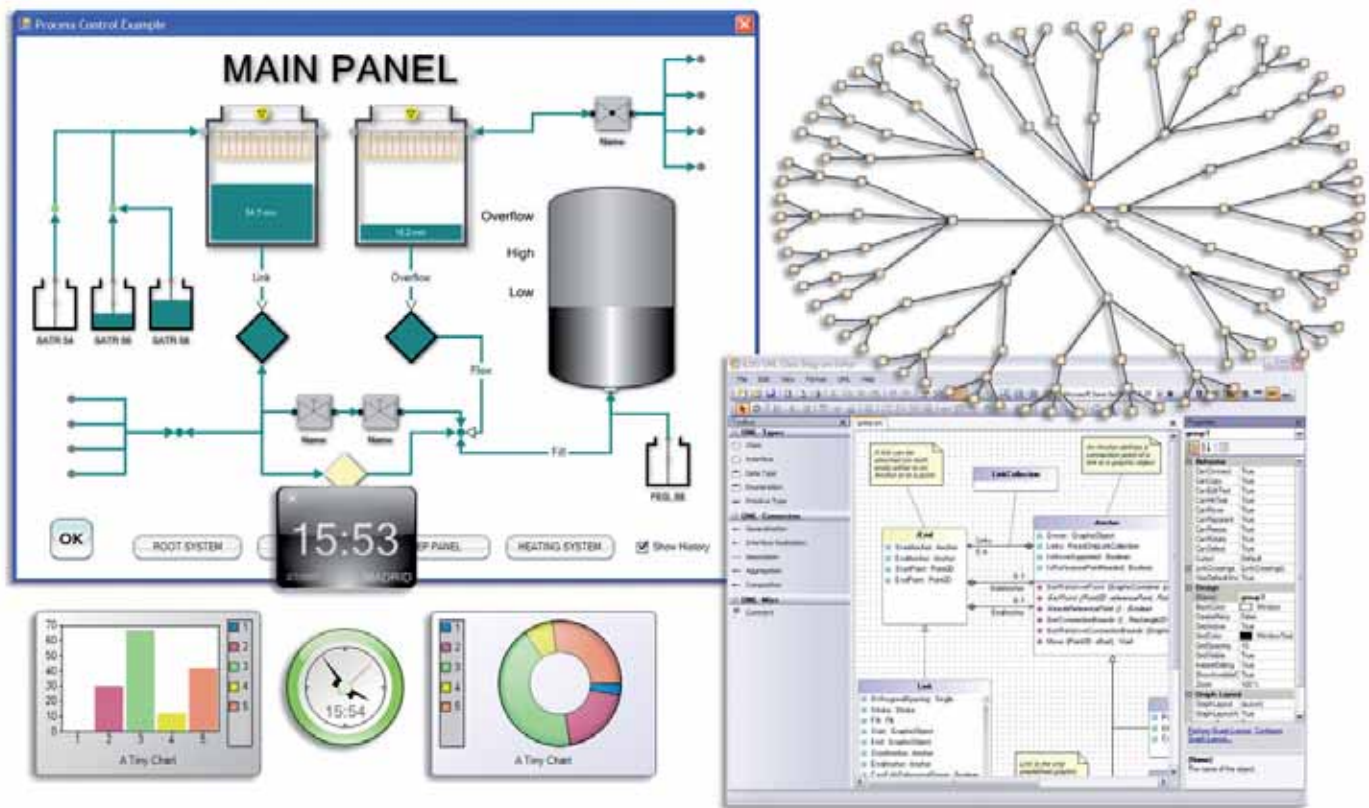
Visual C++ 10 Bibliothèques : le C++ est loin d'être mort et encore moins les classes MFC. Il est déjà prévu un alignement de l'interface et des objets d'interfaces avec les futurs Office et Windows (avec dans ce cas, l'intégration des nouvelles fonctions systèmes), l'interopérabilité entre le code natif et managé sera une des priorités.

Oslo (à partir de 2009) : le projet Oslo est particulièrement ambitieux pour Microsoft et impacte directement le futur du développement. L'ambition est de bâtir une plate-forme complète pour la SOA, les Softwares + Services, les applications composites et bien entendu les applications actuelles. Oslo définit 5 piliers : Biztalk Server, Biztalk Services, Visual Studio, System Center et .Net 4.0. Mais surtout, il s'agit de mettre en œuvre les notions d'usines à logiciels. Pour ce faire et fédérer les outils de développements, serveurs, etc., on disposera d'un référentiel commun. On pourra aussi consommer les applications Oslo dans les services Live, pareillement dans Office qui devient un conteneur Oslo. Les premières Bêta apparaîtront courant 2008. Nous aurons l'occasion cette année de revenir sur ces nombreux projets et outils !

■ François Tonic

NOUVEAU

ILOG Diagrammer for .NET



- Diagrammes et tableaux de bord pour tous les types d'applications graphiques
- Algorithmes d'arrangement automatique de graphiques et de diagrammes
- Éditeurs graphiques prêts à être intégrés
- Intégration totale avec Microsoft Visual Studio
- Création d'applications riches ou Web avec Ajax
- Tableaux de bord avec animation en temps réel

<http://diagrammerfor.net.ilog.com>

Microsoft
CERTIFIED
Partner

ISV Software Solutions

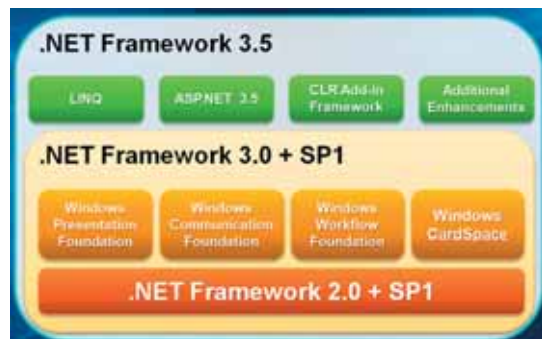
Optimized for
Visual Studio .NET



Changing the rules of business™

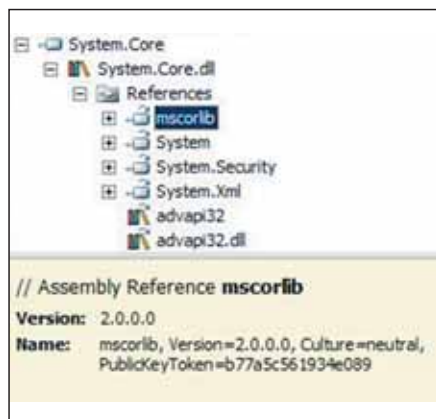
Comment et pourquoi migrer les projets .Net vers .Net 3.5 ?

Le framework 3.5 est sorti il y a quelques mois. Mais on a encore peu évoqué la problématique de migration des projets, des codes .Net 1.x / 2.x, voire 3.0, vers le nouveau framework. Qu'en est-il réellement ? Faisons ensemble un premier tour d'horizon.



Rappelons tout d'abord ce qu'est .Net 3.5. Une piqure de rappel n'est jamais inutile ! Voici le package du .NET Fx 3.5 : Construit sur la base des versions précédentes du .NET Fx (.NET 3.0 + SP1 / .NET 2.0 + SP1), le .NET Fx n'est donc qu'une nouvelle version de .NET apportant de nombreuses améliorations dont les principales sont :

- LINQ
 - ASP.NET 3.5
 - CLR Add-IN framework (blog de Jack Gudenauf - CLR Software Architect)
 - ... et d'autres améliorations considérées comme mineures face aux trois précédentes.
- Notons que .Net 2 reste le cœur de l'univers .Net...

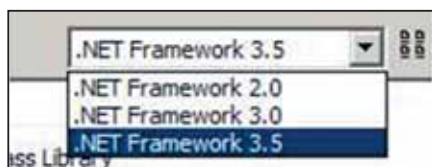


La System.Core est une librairie du .NET 3.5. Observez sa mscorlib dont la version est bien 2.0.0.0. Retenez bien que la CLR n'a pas subi de modifications, changements majeurs depuis la v2. Microsoft a surtout rajouté des librairies autour du cœur. .Net est d'une très grande modularité, souplesse. Enfin, .Net repose sur deux éléments majeurs : la CLR (Common Language Runtime) et les BCL (Base Class Libraries).

Quelles implications de migration de mes projets .NET 2.0 vers 3.5 ?

Comme nous l'avons dit plus haut, .Net 3.0 et 3.5 ne modifient pas la CLR 2.0. Il serait donc facile de conclure que l'exploitation des nouveautés des versions 3.x est simple et que l'impact sur le code est mineur (pour la modification). Tout n'est jamais simple...

Tout d'abord, il faut savoir que l'on n'a pas besoin de VS 2008. On peut référencer les librairies .Net 3.5 dans VS 2005 (nous ne parlons pas d'avoir les templates projets dans VS 2005). Mais, VS 2008 est l'IDE de référence pour coder en .Net 3.5. Et le multi cible (multi targeting) est tout de même un plus pour le développeur .Net.



Ainsi lors de la migration, vous aurez toujours à votre disposition l'outil d'assistance de migration de vos projets (existant déjà dans Visual Studio 2005) et adapté à cette nouvelle version.

Quelles implications de migration de mes projets .NET 1.x vers 3.5 ?

C'est une question récurrente. Que se passe-t-il pour mes codes actuels avec la nouvelle version... Disons le tout de suite : la migration d'un projet .NET de type 1.x devra préalablement subir une migration vers une projet de type .NET 2.0 afin de suivre son évolution vers la version 3.5 (ces étapes ne devant pas être spécialement disjointes). Ce n'est pas une réelle nouveauté. Nous l'avons écrit dans Programmez. Bref si vous êtes encore en v1.x, l'étape 2.0 est obligatoire.

Les scénarios possibles

Quelques scénarios d'évolution vers VS2008 - .NET 3.5 :

- Vous avez Visual Studio 2008 sur votre machine : là, pas de question à se poser, on y va !
- Vous avez Visual Studio 2005 sur votre machine, vérifiez votre configuration :
 - .NET Framework 2.0 + SP1 : passez au .NET 3.0, puis migrez en 3.5
 - .NET Framework 3.0 + SP1 : passez au .NET 3.5
- Vous avez Visual Studio 2002 ou 2003 : passez directement à Visual Studio 2008
- Vous n'avez pas Visual Studio : Installez directement Visual Studio 2008 ... vous avez toujours l'occasion de passer à Visual Studio Express qui est gratuit : <http://www.microsoft.com/express/default.aspx>

La bonne nouvelle est que le package du .NET 3.5 devrait se charger de vérifier votre configuration et n'installer que les packages nécessaires vers le .NET Fx 3.5. Les modifications devraient être mineures mais il y a toujours des cas particuliers. Donc, à vous de procéder à des tests avant de migrer définitivement en 3.5 et en VS 2008. Là, l'utilisation d'un environnement virtuel s'avère utile, sauf si vous avez un multi boot ou une machine de tests... Si vous avez installé des pré-versions de .Net 3.5, VS 2008, désinstallez-les avant !

■ Grégory Renard

CTO Wygwan

Microsoft Regional Director - Microsoft MVP (Most Valuable Professional)

INETA Lead Belgium

<http://www.wygwan.com/> - <http://blogs.developpeur.org/redo/> - WygwanTV

Adapté par François Tonic

La gamme Visual Studio 2008 Express

Découvrez tout ce qu'il faut savoir sur la nouvelle gamme Express 2008 dérivée de la Visual Studio 2008, sortie en novembre dernier.

La gamme de produits Visual Studio Express est un ensemble d'environnements de développement, entièrement gratuits, développés par Microsoft. Ils ne sont que des versions allégées de Visual Studio 2008, l'outil de développement phare de Microsoft.

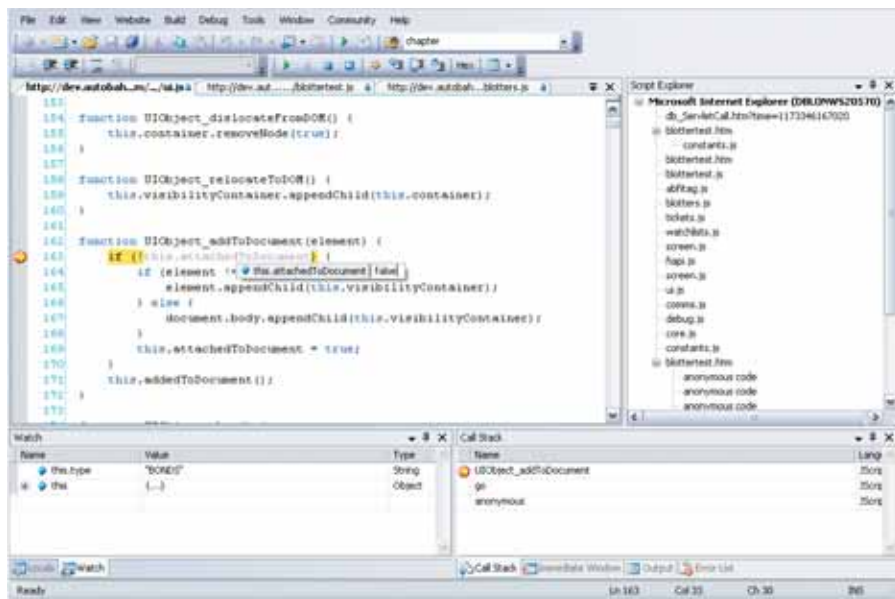
A l'heure actuelle, la gamme Express se compose de 5 produits :

- Visual Basic 2008 Express Edition
- Visual C# 2008 Express Edition
- Visual C++ 2008 Express Edition
- Visual Web Developer 2008 Express edition
- SQL Server 2008 Express Edition (disponible courant 2008).

Vous l'aurez compris, chacune des versions Express cible un langage ou une technologie différente. On ne peut donc pas vraiment dire que ces versions sont concurrentes mais plutôt complémentaires les unes par rapport aux autres. Attention, le concept de ces versions Express n'est pas nouveau chez Microsoft : en effet, la gamme de produits Visual Studio 2005 incluait déjà ces versions et, devant leur succès, Microsoft n'a eu d'autre choix que de les inclure à nouveau dans la nouvelle gamme de produits Visual Studio.

Quelle utilité ?

L'objectif de Microsoft, au travers de ces versions Express, est de fournir un IDE (Integrated Development Environment) simplifié, facile à utiliser et à prendre en main. En effet, lorsque l'on débute en programmation, et que l'on se trouve face à un environnement de développement tel que Visual Studio, on peut vite se retrouver perdu, ne plus savoir où cliquer, où se trouve la boîte de dialogue qui nous intéresse, etc. Avec les versions Express, ces problèmes sont limités de par le fait qu'il s'agit de versions simplifiées. Mais cela ne veut pas dire que les fonctionnalités sont différentes....



Visual Web Developer 2008 Express edition.

A qui s'adresse-t-on ?

Une question qui revient souvent, lorsque l'on parle des versions Express, est de savoir à qui elles s'adressent. De par leurs objectifs (permettre le développement d'applications .NET), les versions Express de Visual Studio 2008 s'adressent à tous les développeurs Windows (VB, C++), c'est une première chose. Ensuite, la gratuité des produits en fait des environnements de développements parfaitement adaptés pour des étudiants ou des développeurs amateurs. N'importe qui peut les utiliser et, dans bien des cas, il s'agit de produits qui conviennent et qui évitent de sortir toute la panoplie : quel intérêt de lancer Visual Studio Team Suite lorsque l'on doit juste réaliser une application console afin de faire des tests ?

Les nouveautés

Si le concept Express n'est pas nouveau. La gamme 2008 et l'arrivée du framework 3.5 font évoluer les outils. Ainsi, on retrouve un designer WPF et le support de C# 3 (expressions lambda, requêtes LINQ, types anonymes, inférence de type grâce au mot-clé var, etc.) et VB 9.

Autre grande nouveauté : le support de ce que l'on appelle le "multi-targeting". A quoi est-ce que cela correspond ? Tout simplement à la possibilité de pouvoir sélectionner quelle version du Framework .NET votre application devra utiliser (.NET 2, 3 ou 3.5).

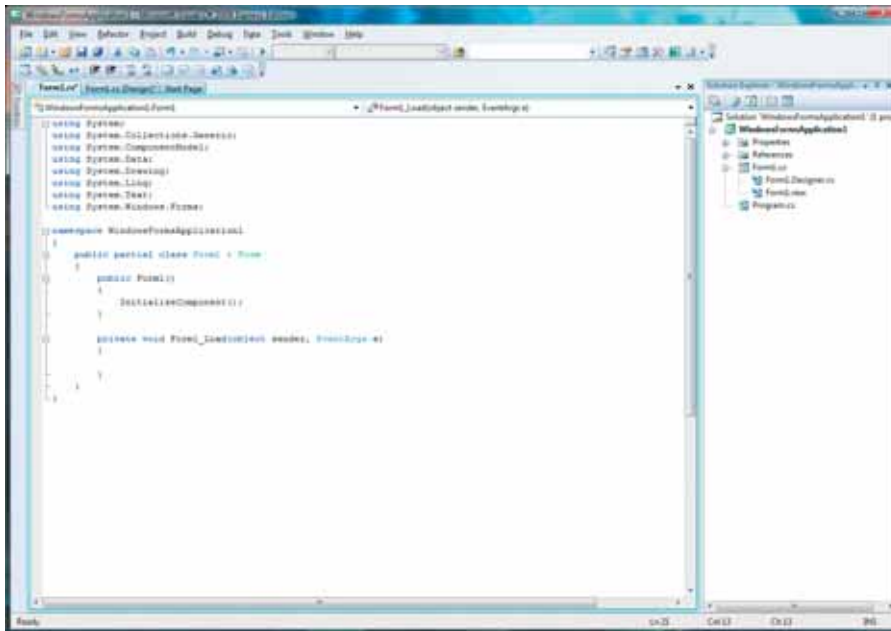
Bien entendu, les développeurs Web n'ont pas été oubliés car Visual Studio Web Developer Express comporte, lui aussi, des nouveautés intéressantes comme un concepteur HTML et un support des CSS largement amélioré ainsi que l'Intellisense et le débogage du code Javascript !

Avantages et inconvénients

Comme vous pouvez vous en douter, ces versions Express ont des avantages et des inconvénients, ce que nous allons voir tout de suite.

Les versions VB.NET et C# incluent, toutes les deux, le designer WPF (Windows Presentation Foundation) nommé "Cider". De plus, l'Intellisense a été retravaillée et on dispose maintenant d'une meilleure correction de la syntaxe. Pour faire simple, tout ce qui était disponible dans les versions Express 2005 l'est avec la gamme 2008, plus les nouveautés liées au Framework .NET 3.5.

Cependant, ces deux produits possèdent des inconvénients qui risquent d'en décevoir plus d'un. Premièrement, il n'y a pas de support, au niveau de l'environnement de développement, pour des bases de données autres que SQL Server Express ou Access. Il n'est pas possible de développer des applications pour périphériques mobiles (PDA, Smartphone, etc.) et Crystal Reports n'est, malheureusement, pas disponible.



Visual C# 2008 Express Edition.

Dans la série des inconvénients, il y a également le fait qu'on retrouve moins de modèles de projets, ainsi que des options limitées pour tout ce qui est débogage et points d'arrêt.

En effet, dans la version C# Express, la liste des points d'arrêt a été supprimée : les utilisateurs ne peuvent donc plus qu'ajouter ou supprimer des " breakpoints ". Enfin, toujours dans la version C#, certaines méthodes de " refactoring " ont été supprimées. La raison officielle de ce changement est que cela permet de simplifier, au mieux, l'expérience utilisateur lors de l'utilisation de C# Express.

Enfin, une chose pourra perturber les développeurs utilisant VB.NET Express : il n'est pas possible de développer des applications ASP.NET. Vous devez utiliser Web Developer Express (ou une version plus complète de Visual Studio) pour cela.

En voyant tous ces inconvénients, vous pourriez vous demander alors quelles raisons vous auriez d'utiliser l'une ou l'autre de ces versions.

L'une des raisons qui me vient à l'esprit est la rapidité de l'environnement. En effet, alors qu'une version Team Suite de Visual Studio est lourde à charger et à utiliser, les versions Express sont légères et très rapides. Du coup, si vous avez besoin de faire tourner un Visual Studio sur une machine qui n'est pas de toute dernière génération, il y a fort à parier que l'une ou l'autre (ou plusieurs) de ces versions Express devraient faire votre bonheur.

Sinon, pour tout le reste, si vous avez juste besoin de faire un simple développement Win-

dows/.NET et que vous n'avez pas besoin de vous occuper de toute la partie architecture, tests, couverture de code, etc. alors ces versions Express sont faites pour vous !

Je tiens cependant à souligner un autre point important lié aux versions Express. La licence d'utilisation vous permet d'utiliser et de développer des logiciels, que vous pourrez (si vous le souhaitez) revendre. Cependant, cette même licence vous interdit formellement d'étendre (au moyen d'add-in), votre environnement de développement. Ainsi, si vous savez que vous devrez, au cours de votre projet, utiliser des add-in tels que celui pour le site Plnvoke.net, TestDriver.Net ou GhostDoc, alors il vaut mieux penser à une autre solution que ces versions Express.

Les différents projets autour des versions Express

Il ne faut pas croire que sous prétexte qu'il s'agit de versions allégées, il ne doit rien y avoir autour. Ainsi, bon nombre de projets gravitent autour de cette gamme de produits. Par exemple, si vous utilisez Visual Web Developer Express, il n'y aura rien de surprenant à ce que vous utilisiez aussi **Microsoft Popfly** (<http://www.microsoft.com/express/popfly/>), les outils **Windows Live** (<http://go.microsoft.com/?linkid=7653524>), le " **Facebook Developer Kit** " (<http://go.microsoft.com/?linkid=7653523>) ou bien encore les **Windows Live Quick Applications** (<http://go.microsoft.com/?linkid=7716695>).

Si vous préférez le développement Windows, sachez qu'il n'a pas été oublié et qu'un certain nombre de projets y sont consacrés. Le kit de développement " **Coding4Fun** " (<http://www.microsoft.com/express/samples/c4fdevkit/default.aspx>) est une collection de wrappers, en code managé, utilisés pour accéder aux API de Windows Vista directement depuis la boîte à outils de Visual Studio.

Le " **Vista P2P Developer Kit** " (<http://www.microsoft.com/express/samples/c4fp2p/Default.aspx>) vous sera particulièrement utile si vous désirez intégrer des fonctionnalités de Peer-To-Peer dans vos applications.

Enfin, le " **Facebook Developer Kit** " (<http://www.microsoft.com/express/samples/facebook/default.aspx>) est également disponible. Libre à vous alors d'intégrer un peu de Facebook dans votre application WPF ou Windows-Forms ! Si vous êtes plutôt développeur C++, vous avez de quoi être heureux avec le " **Game Creators GDK** " (<http://go.microsoft.com/?linkid=7634886>), Visual Studio C++ 2008 Express et XNA Game Studio Express (<http://go.microsoft.com/?linkid=7694047>) qui vous permettront de développer, à loisir, vos propres jeux 2D et 3D.



Généré par XNA.

Conclusion

Les versions Express de Visual Studio sont des outils vraiment performants qui, de par leur gratuité, feront le bonheur du plus grand nombre. Microsoft semble encore avoir eu une bonne idée en permettant à tout un chacun d'utiliser ces outils sans frais : un bon moyen de continuer à gagner des parts de marché auprès des développeurs qui ne se sont pas encore mis à .NET !

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



NetAdvantage® for .NET



NetAdvantage for ASP.NET

Des composants haute performance pour interfaces utilisateur. Une expérience supérieure pour vos utilisateurs.

Des composants haute performance pour interfaces utilisateur

Recevez une boîte à outils complète pour multi-plateformes, en influençant même les applications les plus exigeantes incluant des grilles riches et de haute performance pour Windows Forms et ASP.NET

Compatibilité Visual Studio®

Utilisez la version de Visual Studio qui vous convient - VS 2005 ou VS 2008 avec un support complet pour NetAdvantage

Créez des tableaux de bord professionnels

Présentez des graphiques et gauges haute fidélité pour ASP.NET et Windows Forms afin d'obtenir une présentation limpide des Key Performance Indicators

Design Once, Style Everywhere

Donnez instantanément un style professionnel à vos applications créées avec NetAdvantage pour imiter Office® 2007, Vista® ou bien créez et appliquez vos normes corporatives à travers les contrôles et vos applications

Accélérez vos applications web

Des composants rapides et légers, basés sur notre nouveau framework AJAX Aikido™, sont essentiels pour des applications agiles pour Web 2.0 et SharePoint® Web Parts

Donnez du pouvoir à vos utilisateurs

Délivrez des interfaces utilisateur riches et productives à vos clients, pour ASP.NET et Windows Forms, également disponible pour WPF ou pour JSF

Pour de plus amples informations:
infragistics.com/dotnet

sales-europe@infragistics.com

N° Vert 0800 667 307

Infragistics®
Powering The Presentation Layer

Infragistics Sales 800 231 8588

Infragistics Europe Sales +44 (0) 800 298 9055

WINDOWS FORMS

ASP.NET

WPF

JSF

grids scheduling

charting

toolbars navigation

menus listbars

trees

tabs

explorer bars

editors

& more

Toutes les techniques d'extensibilité de Visual Studio 2008

Enrichir Visual Studio en y incorporant ses propres outils, c'est donner la possibilité à chaque développeur de personnaliser son environnement de développement pour être le plus efficace possible. Les techniques d'extensibilité présentes dans Visual Studio 2005 permettaient déjà de le faire, mais Visual Studio 2008 va plus loin en améliorant ces techniques et en offrant aussi la possibilité de construire un environnement complètement spécifique tout en bénéficiant des fondations de Visual Studio.

L'histoire du Shell de Visual Studio remonte à plus de 10 ans, aux années 95/97. Elle prend racine dans la nécessité qu'avait Microsoft de rationaliser ses environnements de développement avec la même fondation applicative. Version après version, les différents environnements (VC, VB, VI...) se sont unifiés pour donner Visual Studio tel que nous le connaissons maintenant depuis 2001. La notion d'extensibilité, méthode d'enrichissement de Visual Studio par des fonctionnalités tierces, a toujours existé. Auparavant payant pour les versions 2001 et 2003, l'accès au SDK s'est démocratisé à partir de Visual Studio 2005, pour enfin prendre son plein essor avec la version 2008 qui apporte la possibilité de s'approprier et redistribuer le Shell.



Visual Studio 2008 Shell

Visual Studio 2008 Shell, c'est la possibilité de construire ses propres outils de développements et de les distribuer facilement. Auparavant, pour installer des outils de développement tiers ou des extensions basées sur le modèle d'extensibilité de Visual Studio, il fallait nécessairement avoir installé Visual Studio Professionnel ou supérieur, ou obtenir une licence de la version " Partner Premium Edition " qui n'était pas vraiment donnée. Avec le mode Shell, Microsoft donne la possibilité aux développeurs d'extensions de pouvoir les redistribuer gratuitement et sans qu'il y ait nécessairement comme préalable Visual Studio installé. Le Shell supporte deux modes :

- Le **mode intégré** : recommandé pour les langages de programmation, il permet d'intégrer automatiquement les extensions construites avec ce mode à n'importe quelle édition de Visual Studio installée sur la même machine.
- Le **mode isolé** : recommandé pour les outils spécialisés, il s'installe et s'exécute " à côté " de Visual Studio, quelle que soit son édition. Ce mode vous permettra de personnaliser d'avantage le Shell que le mode intégré, par exemple, de modifier la fenêtre de bienvenue (" Splash Screen "). A noter dans ce cas, que la seule " trace " visible qu'il s'agit de Visual Studio est la présence d'un message " Powered by Visual Studio " en bas à droite de la fenêtre de bienvenue.

La figure suivante montre un exemple d'utilisation du Shell Visual Studio en mode isolé. Cet exemple, appelé StoryBoard, est disponible sur CodePlex et propose un environnement complet et totalement person-

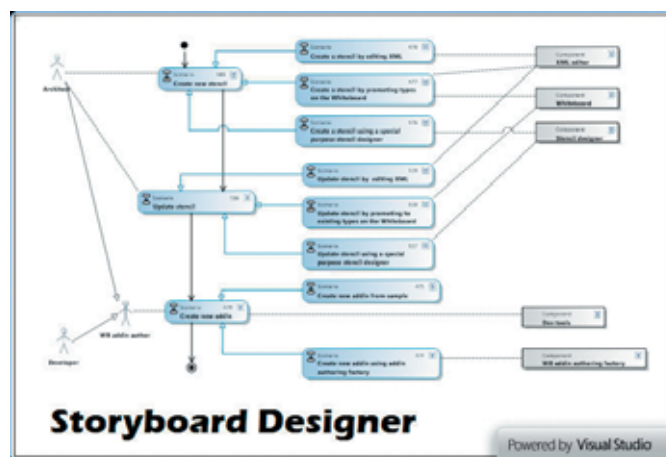


Figure 1 – La fenêtre de bienvenue de l'application StoryBoard, environnement de développement construit sur Visual Studio Shell en mode isolé.

nalisé composé de DSL (Domain Specific Language), d'un type de projet et de templates spécifiques. Téléchargement : <http://www.codeplex.com/storyboarddesigner>.

Quel que soit le mode que vous choisirez pour développer des extensions à Visual Studio 2008, le Shell vous apporte un ensemble de fonctionnalités homogènes, vous permettant d'enrichir l'expérience du

développeur de diverses manières : intégration d'un nouveau langage de programmation, un environnement de développement propre à votre métier, un générateur de code intégré et bien d'autres scénarii.

La figure ci-contre montre le positionnement des différentes éditions de Visual Studio ainsi que le Shell en mode intégré ou isolé.



Figure 2 – Les différentes versions de Visual Studio et du Shell.

Finalement, pour redistribuer votre application ou vos outils construits à l'aide du Shell et suivant le mode, vous aurez le choix du :

- **Package de redistribution de Visual Studio 2008 Shell (mode isolé) :** Ce package fait 300 Mo et inclut .NET Framework 3.5 ainsi que les composants Visual Studio Shell. Pour redistribuer votre Shell, vous aurez besoin d'une clé de chargement (Shell Load Key) fournie par Microsoft.
- **Package de redistribution de Visual Studio 2008 Shell (mode intégré) :** Ce package fait 300 Mo et inclut .NET Framework 3.5 ainsi que les composants Visual Studio Shell.

Toutes les informations sur Visual Studio 2008 Shell sont accessibles sur MSDN : <http://msdn2.microsoft.com/en-us/vstudio/bb510103.aspx>.

Enrichir & étendre Visual Studio

Le SDK de Visual Studio est un préalable au développement d'extensions quelle que soit l'édition ciblée (Visual Studio, Visual Studio Shell mode intégré et/ou isolé). L'utilisation du SDK ne pourra se faire qu'à partir de l'édition Standard de Visual Studio. Le " Visual Studio 2008 SDK 1.0 " est téléchargeable à partir du portail de l'extensibilité de Visual Studio sur MSDN : <http://msdn2.microsoft.com/en-us/vstudio/aa700819.aspx>. Outre l'aide précieuse qu'il apporte, le SDK est nécessaire dans le développement de " packages ".

Voilà, nous sommes maintenant parés pour étendre Visual Studio. Mais quelles sont les différentes manières d'enrichir l'environnement de développement ? Il en existe 3 :

- Les macros,
- Les add-in,
- Les packages, aussi appelés VSPackages.

Les macros

Les macros sont la manière la plus simple d'étendre Visual Studio via son modèle objet. Les macros sont très pratiques pour automatiser des tâches répétitives. La plupart du temps, le développeur utilise les macros pour reformater le code ou insérer du code répétitif par exemple. Visual Studio fournit un enregistreur de macro qui facilite grandement le développement ou la modification de macros.



Figure 3 – L'explorateur de macros.

L'ensemble des fonctionnalités relatives aux macros est accessible par le menu **Tools > Macros**. Il est possible de voir l'ensemble des macros disponibles (**Macro Explorer**), d'en enregistrer, d'en éditer ou de les exécuter. L'enregistreur est le moyen le plus simple pour générer le code correspondant à la tâche que l'on souhaite exécuter et ensuite de modifier ce code pour l'adapter. De plus, l'enregistreur est un outil qui permettra au développeur de mieux appréhender de manière générale le modèle objet de Visual Studio. La figure ci-contre montre l'explorateur de macros.

Les macros sont exclusivement écrites en VB.NET et n'ont pas accès aux fonctions internes de Visual Studio. Le développeur doit fournir son code pour redistribuer une macro car celui-ci n'est pas compilé et reste donc modifiable.

Les add-in

Les add-in, parfois appelés plug-in, permettent eux aussi d'avoir accès au modèle objet de Visual Studio dans le même ordre d'idée que les macros. Les add-in sont écrits dans n'importe quel langage de la plateforme .NET et sont compilés, ce qui permet de dépasser la limitation des macros pour la redistribution. Lorsque l'on développe un add-in, nous avons accès à un objet " racine " appelé DTE/DTE2 qui permet d'accéder ensuite à tout le modèle objet de Visual Studio. Vous pouvez éditer une macro comme expliqué précédemment pour en comprendre son utilisation. Le modèle objet est visible sous la forme d'un poster [http://msdn2.microsoft.com/en-us/library/za2b25t3\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/za2b25t3(VS.80).aspx).

Il existe beaucoup d'add-in disponibles en téléchargement sur le Web et qui permettent de faire d'innombrables choses : des add-in d'intégration d'outils externes à Visual Studio, des add-in de génération de code, des add-in d'optimisation du code, ... Des sites Web référencent la plupart d'entre eux et notamment, le site Visual Studio Hacks : <http://www.visualstudiohacks.com/type/addins>.

Comment créer son propre add-in dans Visual Studio 2008 ? Rien de plus simple :

1. **File > New Project > Other Project Types > Extensibility > Visual Studio Add-in.** Un assistant va ensuite nous aider à créer le squelette de notre add-in.
2. Cliquer sur le bouton " Next " sur l'écran d'accueil de l'assistant.
3. Sélectionner les options nécessaires à votre add-in. Dans cet exemple :
 - a. "Create an Add-In using Visual C#".
 - b. Par défaut, support de toutes les versions depuis Microsoft Visual Studio 2005 – Petit détail au passage : la plupart des techniques d'extensibilité de Visual Studio 2008 sont compatibles binaire et de manière ascendante avec la version 2005.
 - c. Nommez votre add-in.
 - d. Cochez toutes les options disponibles pour les add-in.
 - e. Ajoutez une boîte d'information.
4. La validation de l'assistant entraîne la création de l'add-in avec le code correspondant aux options.

Le code de la méthode **Exec** est modifié de la manière suivante pour faire apparaître un message :

```
if(executeOption == vsCommandExecOption.vsCommandExecOption
DoDefault)
{
    if(commandName == "MyHelloWorldAddin.Connect.MyHelloWorld
Addin")
    {
        MessageBox.Show("Hello World");
        handled = true;
        return;
    }
}
```

Pour tester notre add-in, il suffit juste d'exécuter notre projet en mode debug (F5 ou menu **Debug > Start Debugging**), ce qui a pour effet de



Figure 4 – Exemple d'add-in

lancer un second Visual Studio. Par défaut, votre add-in se trouvera dans le menu **Tools** si vous n'avez pas modifié le code de la méthode **OnConnection**.

Il existe des exemples plus complets et notamment l'exemple d'implémentation d'un add-in qui convertit le code C# vers Visual Basic (<http://msdn.microsoft.com/msdnmag/issues/06/02/PasteAs/>).

Pour contrôler le chargement des add-in Visual Studio, il existe un gestionnaire d'add-in accessible par le menu **Tools > Add-in Manager**. Vous pouvez ensuite redistribuer votre add-in en créant un projet Setup pour générer une installation de type MSI.

Les VSPackages

Les VSPackages représentent le mécanisme de plus bas niveau disponible au développeur pour étendre Visual Studio. Plus complexes, mais plus puissants, ils permettent d'exploiter les services de Visual Studio comme le font les packages C# ou VB.NET ; En fait les éditeurs, barres de commandes, etc. que nous voyons dans Visual Studio sont eux-mêmes des VSPackages. Auparavant, ce mode d'extensibilité n'était accessible qu'au développeur C++ mais depuis la version 2005 il est possible d'interopérer avec les classes natives de Visual Studio à partir des langages .NET, soit en utilisant les classes d'interopérabilité, soit en utilisant le **Managed Package Framework (MPF)**. Tout comme pour la création d'add-in, Visual Studio fournit un assistant de création de Package. Dans ce cas, le type de projet à utiliser est " **Visual Studio Integration Package** ".

Un VSPackage vous permet d'ajouter des informations sur votre outil dans la fenêtre d'accueil de Visual Studio, d'implémenter un nouveau langage de programmation, de modifier le comportement d'éditeurs de code existants, etc. Tout ceci est possible grâce aux fonctionnalités mises à disposition par le Shell et les autres VSPackages, comme :

- **Les Services** : Les VSPackages sont capables de consommer et d'être consommés par d'autres VSPackages. Pour ce faire, un VSPackage doit publier ou consommer un service, sorte de contrat représentant un ensemble d'interfaces utilisables par les VSPackages. Visual Studio 2008 lui-même est un ensemble de VSPackages interconnectés par les services. La liste complète des services utilisables est disponible sur MSDN : [http://msdn2.microsoft.com/en-us/library/bb165789\(VS.90\).aspx](http://msdn2.microsoft.com/en-us/library/bb165789(VS.90).aspx).
- **Managed Language Services** : Ce service permet d'étendre ou d'implémenter un nouveau langage de programmation. Ce service est utilisable par les langages .NET, mais le développeur C++ pourra utiliser son équivalent natif (Language Service).

- **Tool Window** : Ce sont les fenêtres intégrées à l'IDE qui permettent d'afficher de l'information. Exemple : Toolbox, Solution Explorer, Properties...
 - **Document Windows** : Ce sont les fenêtres MDI de l'IDE. Ces fenêtres permettent généralement d'éditer du code, de concevoir des interfaces.
 - **Editor** : C'est un VSPackage qui permet d'implémenter la logique d'un éditeur (texte, code...).
 - **Menus et barres d'outils** : Ils permettent aux développeurs d'accéder aux fonctionnalités des VSPackages par des éléments d'interfaces intégrés à l'IDE. Ce sont généralement des commandes implémentées par les VSPackages.
 - **Projects** : Ce sont des conteneurs utilisés pour organiser les fichiers composant un projet. Vous pouvez tout à fait implémenter un nouveau type de projet.
 - **Templates** : Ce sont des projets ou éléments de projet préconfigurés. Vous pouvez créer vos propres templates et les distribuer.
 - Bien d'autres fonctions encore comme, le contrôle de code source, l'intégration à l'aide, les outils DSL, les outils de tests de l'édition pour les testeurs, l'extensibilité du concepteur de données (DDEX), le SDK de débogage, ou encore le modèle de définition de système (SDM – System Definition Model) pour la conception, la configuration et l'évaluation du déploiement d'applications sur des systèmes cibles...
- Pour redistribuer un VSPackage, vous aurez besoin d'une clé de chargement appelée Package Load Key (PLK). Cette clé vous sera fournie par Microsoft via l'inscription au programme partenaire VSIP accessible sur le site Web suivant : <http://www.vsipmembers.com/Anonymous/Default.aspx>. Il existe plusieurs exemples d'extensibilité de Visual Studio par de nouveaux types de projets ou l'enrichissement d'existants, de nouveaux outils, etc. Voici 2 exemples de nouveautés fournies par Microsoft qui viennent enrichir Visual Studio pour développer des applications Web : Popfly et Volta, une technologie alpha pour développer des applications Web multi-tiers (Silverlight, Javascript, .NET...) :
- Microsoft Popfly & Visual Studio : <http://www.popfly.ms/Overview/Explorer.aspx>
 - Microsoft Live labs Volta & Visual Studio: <http://labs.live.com/volta/>

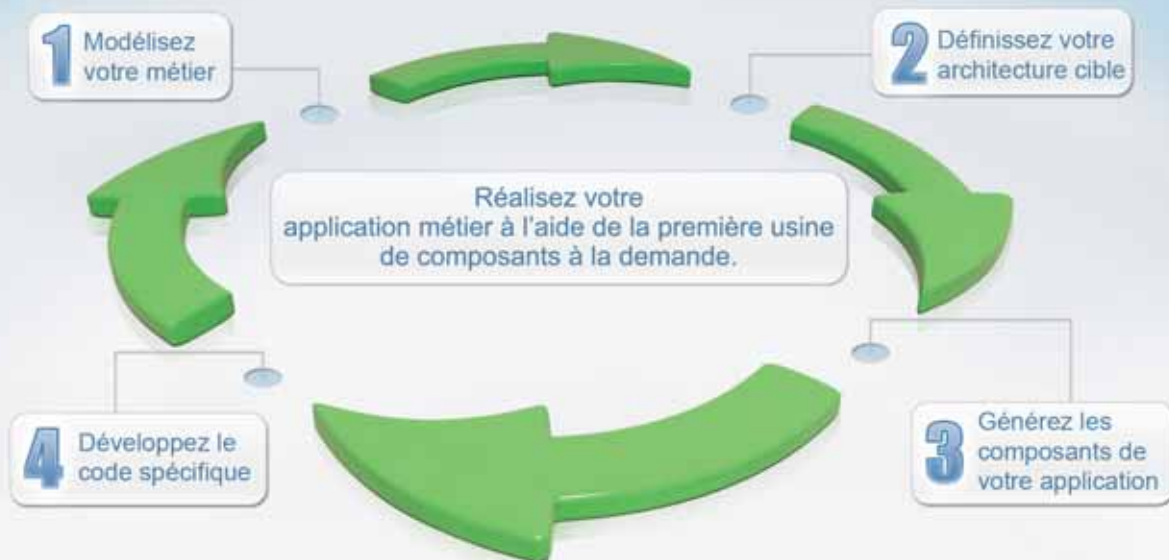
Conclusion

Avec l'arrivée de la version 2008 de Visual Studio, ce sont de nouvelles capacités d'extensibilité qui s'offrent aux développeurs. L'objectif des extensions (add-in, VSPackages...) de Visual Studio est généralement d'améliorer la productivité et l'efficacité des développeurs, d'enrichir les possibilités avec, par exemple, de nouveaux langages de programmation (exemple de IronPython) ou de nouveaux types de projets. C'est tout un écosystème (communauté VSX) qui se met en place autour du métier de développeur en lui apportant un nouvel aspect et un nouvel intérêt et qui permet à chacun de se forger les outils qui lui sont adaptés.



■ Frédéric QUEUDRET

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



CodeFluent.com

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



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

0€

CodeFluent

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



Prix public : ~~2490€ht~~

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

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



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

Softfluent recrute !

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



Quoi de neuf dans PowerShell 2.0 ?

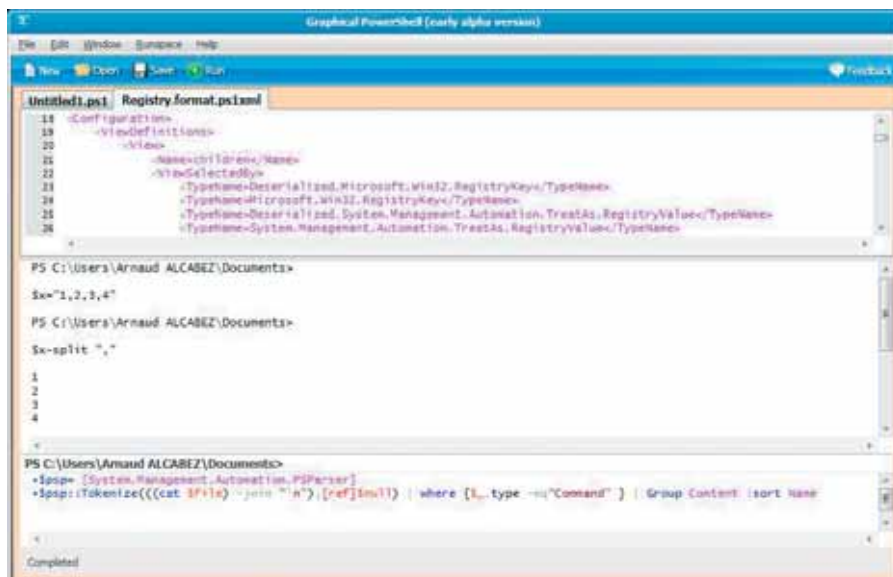
Le 6 Novembre 2007, lors des TechEd européens à Barcelone, Jeffrey Snoover a présenté les derniers développements de l'environnement de scripts PowerShell. La version 2.0 apporte pas mal de nouveautés.

PowerShell est un langage de script extensible et un environnement interactif offrant un traitement des tâches orienté commande. En ce sens, PowerShell facilite les tâches d'administration, comme la gestion du paramétrage des composants système imprimante, réseaux, IE, la création de comptes utilisateurs et les opérations associées. Pour fonctionner, PowerShell s'appuie sur la plate-forme .NET, ce qui signifie que tout est objet sous PowerShell.

D'abord confiné dans une longue période de tests sous le nom de Monad, PowerShell a fini par être distribué à partir de 2006 sous forme d'un exécutable indépendant. C'est un pari risqué pour Microsoft de fabriquer un successeur aux batchs MS-DOS et à Windows Scripting Host. Toutefois, PowerShell est au cœur de la stratégie de l'administration des plates-formes Windows. Sa première réelle implémentation a eu lieu avec Exchange Server 2007, la messagerie de Microsoft, où le produit a été complètement repensé pour s'appuyer totalement sur ce langage, au sens où un grand nombre de fonctionnalités ne sont accessibles que par le biais de cette interface. C'est également un revirement de la pensée de Microsoft, puisque pendant des années, la société a défendu l'idée d'un système entièrement graphique.

Au final, PowerShell aura été téléchargé plus de 1,55 millions de fois en un an. Les logiciels Microsoft tels qu'Exchange 2007, MOM 2007, Virtual Machine Manager 2007, Lotus Domino Transporter Suite, Data Protection Manager, Compute Cluster Tool Pack, Windows Server, Windows Backup, Windows Vista, SMS 2003 l'utilisent. Plus de 20 nouveaux partenaires développent des cmdlets (une commande) PowerShell. Désormais, PowerShell fait partie du Common Engineering Criteria, ce qui sous-entend que tous les produits délivrés en 2009 contiendront PowerShell. Bref, pour tous les administrateurs Windows, à partir de 2009, PowerShell deviendra une obligation !

Les communautés de développement sont en train de s'organiser autour de ce langage. Plus de 25 projets Codeplex, dont 2 parmi les 25 plus actifs, et plus de 14 livres ont déjà été



écrits, ainsi que de nombreux outils d'aide au développement autour de PowerShell.

Windows PowerShell 2.0

PowerShell V1 souffre de nombreux défauts. La version présentée par Jeff Snoover a juste-ment pour objectif de les combler. Pour cela, de nombreux feedbacks ont été collectés chez les premiers utilisateurs de PowerShell, et ont été intégrés dans la version présentée en Novembre 2007, sous la forme d'une " Community Technology Preview " (CTP).

Installer Windows PowerShell v2

PowerShell V2 fonctionne sur Windows XP SP2 et les versions supérieures, dont Windows Vista. PowerShell V2 demande à ce que le WS-Management V1.1 soit installé en premier. Ce composant est utilisé pour activer l'exécution de scripts distants. PowerShell V2 remplace complètement la V1.0 et celle-ci doit être désinstallée avant d'installer la V2. La V1 est compatible avec la V2, à l'exception des additions de la V2 : Nouveaux noms de variables, nouvelles fonctionnalités du langage ou des cmdlets. Certains changements peuvent poser des problèmes pour l'exécution de scripts PowerShell V1. Par exemple, deux nouveaux mots clés du langage ont été ajoutés : " Data "

et " Cmdlet ". Si vous avez utilisé ces termes dans vos scripts, dans la V2 ils seront interprétés comme des fautes de syntaxe.

Site : <http://www.microsoft.com/downloads/details.aspx?FamilyID=60deac2b-975b-41e6-9fa0-c2fd6aa6bc89&displaylang=en>

Les nouveautés de PowerShell V2

De nombreuses nouvelles fonctionnalités ont été ajoutées à PowerShell V2, en plus de quelques améliorations de commandes existantes. Dans le cadre du principe de précaution, Microsoft a pris un soin tout particulier à vérifier la meilleure compatibilité possible avec PowerShell V1. Dans la liste des nouveautés, on retiendra particulièrement :

- **PowerShell Remoting** : Windows PowerShell V2 fournit le support pour exécuter les cmdlets et scripts à distance. PowerShell Remoting permet de contrôler un ensemble de machines distantes depuis un seul poste. La gestion des ordinateurs à distance nécessite que PowerShell V2 soit installé sur tous les ordinateurs participant à l'opération. Pour le moment, pour utiliser la fonction d'exécution à distance, il est nécessaire que l'environnement PowerShell soit démarré avec une élévation de privilège de type administrateur. Pour exécuter des

scripts PowerShell à distance sur un Windows Vista, il est nécessaire que le SP1 de Windows Vista soit installé. L'exécution à distance passe par WinRM, le nouveau composant de Microsoft pour autoriser l'exécution de requêtes WMI au travers d'Internet. WinRM est une réponse au problème de l'utilisation de WMI sur des réseaux complexes, qui au lieu d'utiliser les protocoles RPC/DCOM, s'appuie sur le couple SOAP/Web Service WS-Management Protocol. Ainsi, il sera possible de pouvoir exécuter des scripts PowerShell à distance, dans des environnements réseaux très régulés tels que des firewalls ou des NAT.

Scénarios d'utilisation de PowerShell Remoting

Scénario	Objectif
Console PS → Groupe d'ordinateurs (fan-out)	Réduction du TCO au travers d'automatisation sur un grand nombre d'ordinateurs
Mode Interactif	Remplacement de Telnet de manière sécurisée
Groupe d'ordinateurs → Console PS (fan-in)	Automatisation de la délégation d'administration (pour des services hébergés), Utilisation interactive

- **Exécution en arrière plan** : PowerShell V2 définit le concept de travaux d'arrière-plan (PsJob). Un PsJob exécute une commande ou une expression de manière asynchrone et "à l'arrière-plan" sans interaction avec la console. A l'exécution de la commande, le prompt

est redonné immédiatement à l'utilisateur et vous pourrez examiner les résultats du traitement ultérieurement. Vous pouvez exécuter des travaux d'arrière-plan sur un ordinateur local ou un groupe d'ordinateurs à distance.

- **Des outils graphiques** : Une console PowerShell GUI fait son apparition. Il s'agit d'une version très préliminaire qui permet toutefois, sans devoir faire appel à l'un des nombreux produits tiers de développement, de disposer d'une meilleure interface pour le développement de scripts. On peut également noter la présence d'une nouvelle fonction, baptisée **Out-GridView** qui permet de retourner les résultats d'une cmdlet sous la forme d'un tableau graphique. Ainsi, la commande `Get-Process svchost | Out-GridView` donnera le résultat suivant :

```
-Split
$X="1,2,3,4"
$X-split ","

-Join
$X="1","2","3","4"
$X-join ","
```

- **Des outils pour les développeurs** : Désormais, il devient possible de créer directement

des cmdlets en PowerShell à la place de C# ou de VB.NET. On trouve également un mode avancé pour le débogage intégré directement à la console (Breakpoints et Steppable Pipelines), des nouvelles API (Metadata, Paser Tokenizer, PowerShell Hosting). Trois opérateurs font également leur apparition (`-join`, `-split`, et `@ -splatting`). Enfin, un effort a également été fait pour permettre le support multilingue dans le développement PowerShell.

- **Sécurité** : PowerShell V2 permet désormais de fonctionner avec des espaces fermés (Restricted RunSpaces) où on restreint l'exécution de certaines commandes pour des raisons de sécurité. A l'inverse, PowerShell V2 permet également l'impersonnalisation (changement des crédits) pour permettre l'exécution de cmdlets avec d'autres privilèges.

Au final, 24 cmdlets supplémentaires ont été ajoutées à PowerShell V2. De nombreuses fonctionnalités sont prévues dans les prochaines versions, comme la possibilité qu'un script puisse être exécuté de manière événementielle. Il n'existe pas encore de planning prévisionnel quant à la sortie d'une version finale. Gageons que vous serez nombreux à évaluer cette CTP et faire part de vos feedbacks : <http://blogs.msdn.com/powershell/>

■ Arnaud ALCABEZ

Responsable des infrastructures Microsoft, Devoteam Solutions. MVP Exchange Server. Président du Groupe des Utilisateurs Francophones de Microsoft Exchange Server.

Migration d'un projet Web Visual Studio 2005 vers Visual Studio 2008

La migration d'un projet initialement créé avec Visual Studio 2005 vers Visual Studio 2008 est une opération normalement assez aisée : il suffit de suivre les différentes étapes d'un assistant en validant quelques points si nécessaire. Finalement, en dehors du choix du Framework cible (le 2.0, 3.0 ou 3.5), il y a peu de questions à se poser et l'option Finish, disponible dès le premier écran, convient parfaitement. Et pourtant, le processus peut à l'occasion dérailler. En particulier si vous cherchez à migrer un projet Web. Par défaut, l'assistant crée un répertoire de sauvegarde de la solution à migrer sous le répertoire de cette même solution (voir figure).

Une fois le projet migré, lorsque vous lancez la compilation, vous obtenez le message d'erreur suivant :

L'utilisation d'une section inscrite comme allowDefinition='MachineToApplication' au-delà du niveau d'application est une erreur. Cette erreur peut provenir d'un répertoire virtuel non configuré en tant qu'application dans IIS.

En faisant attention à la source de l'erreur, on s'aperçoit que le fichier posant problème est le web.config copié dans le répertoire Backup. Et c'est bien là la source du problème, lors de la compilation, Visual Studio 2008 parcourt tous les répertoires du projet pour traiter tous les fichiers présents. C'est une particularité des



projets Web et cela inclut le répertoire de sauvegarde automatiquement créé. Il y trouve un fichier de configuration qu'il tente d'exploiter. Et c'est à ce moment qu'il tombe sur des directives qui ne peuvent être définies au niveau d'un répertoire (le nœud " authentication " par exemple). La compilation échoue alors avec le message précédent. Mais comme ces fichiers n'apparaissent pas dans la solution, cela peut facilement prêter à confusion.

Le plus simple et le plus rapide est alors de déplacer le répertoire de sauvegarde vers un autre emplacement, de façon à ce qu'il ne soit plus un sous-répertoire du projet à compiler. Et pour les projets suivants à migrer, il faut prendre l'habitude dès le départ de choisir un autre répertoire de sauvegarde que celui proposé par défaut.

■ Patrice Manac'h

Sync Framework : tout pour les données !

En novembre dernier, Microsoft annonçait la disponibilité d'un nouveau framework de développement ainsi que d'un "runtime" baptisé "Sync Framework" (MSF), afin de simplifier le développement de solutions de synchronisation, de roaming et d'applications online ayant la capacité intuitive de fonctionner en mode hors ligne.

Ce nouvel outil, en version CTP1 Refresh, mis à disposition des développeurs .NET semble être une réponse de Microsoft à Google Gears offrant également la possibilité de développer des applications web fonctionnant en mode hors ligne. Ce nouveau Framework vous permettra de développer des scénarios de peer-to-peer, de synchronisation entre tous types de terminaux et services ainsi que le fonctionnement avec des systèmes hétérogènes comme sur les types de Bases de Données, protocoles ou stockages en tout genre. Le Sync Framework, disponible sur le site de Microsoft MSDN, se télécharge en quelques minutes et s'installe très aisément.

Vous y retrouverez deux composants à installer :

- CTP1SyncSetup : Le Framework Sync ainsi que son runtime
- CTP1SyncServicesADO : Les providers et outils ADO

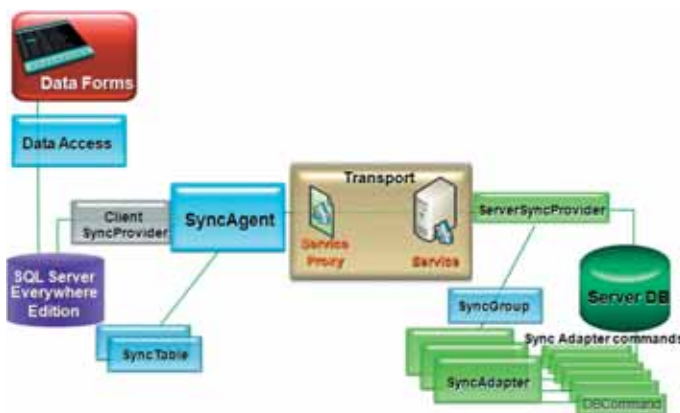
Une fois installé, vous aurez le loisir de parcourir les six bibliothèques mises à disposition, réparties en deux groupes homogènes.

Vous remarquerez rapidement que le MSF repose bien sur la CLR 2.0 du .NET Framework et ne vient ainsi qu'enrichir la liste déjà plus que fournie des bibliothèques du .NET Framework 2.0. Avant de nous lancer dans la découverte d'un exemple concret, tentons de mieux comprendre les objectifs, structures et architectures de ces bibliothèques.

Objectifs de conception du MSF

Sync a ainsi été créé selon des objectifs clés déterminés par l'équipe en charge de son développement :

- **Un Runtime unique**, dans l'optique de base du .NET Framework depuis sa création, celui-ci met à disposition des développeurs souhaitant construire des applications de synchronisation, la capacité de créer des blocs de code génériques, capables de répondre à de nombreuses issues diverses et variées, tant dans la gestion de mémoire cache, de mode hors ligne, de partage de données ou fichiers en tous types que dans le support de scénarios en mode roaming. Ce Runtime contient ainsi la notion d'Agents de synchronisation (SyncAgent) qui se chargera d'exécuter des sessions de Synchronisation faisant, par exemple dans un scénario ADO, transiter des tables (SyncTable) ou groupes de tables (SyncGroup) des données entre deux participants de la synchronisation.
- **Faciliter le flux de contenus** entre les solutions, que celles-ci utilisent ou non différents protocoles ou modes de stockage, en normalisant les métadonnées de synchronisation. Nous entendons par ceci l'exploitation ou la création de providers propres aux différents types de protocoles, bases de données... ex : provider SQL Server ou SQLServer CE ou encore MySQL à terme...
- **La création de scénarios communs** au travers de la simplification du développement actuel de solutions de synchronisations en distribuant des composants de domaines spécifiques comme la synchronisation de bases de données relationnelles, de fichiers, de listes, de terminaux, fichiers de musiques ou vidéos, ...



Composition du MSF

Sync Framework a été conçu selon une architecture en couches et composants afin de permettre aux développeurs de choisir exactement les briques nécessaires à la création de leurs solutions de synchronisation. On retrouve les briques logiques suivantes :

- **Core Sync Runtime** : Cette première partie de l'infrastructure inclut l'ensemble des algorithmes d'exploitation des métadonnées ainsi que les composants de gestion du roaming, du partage, de l'hors-ligne et de la synchronisation de vos applications.
- **Sync Provider Framework** : Cette seconde partie de l'infrastructure du Sync Framework permet d'exposer aisément l'ensemble des données exposées par la plate-forme. Au travers de celle-ci, le Sync Framework offre aux développeurs un modèle de plug-in permettant de configurer les providers de synchronisation ou de créer leurs propres providers.
- **Storage Specific Components** : Cette troisième partie de l'infrastructure facilite le développement rapide de solutions exploitant des systèmes de stockage comme SQLServer, SQL Server CE, NTFS, FAT, ... Chacune de ces briques logiques peuvent être elles-mêmes décortiquées en sous-éléments comme suit :

Core Sync Runtime

Comme mentionné précédemment, celui-ci a pour charge de gérer les services de gestion des métadonnées exploitées par l'ensemble des clients Microsoft Sync Framework. Vous y retrouverez :

- Gestion et représentation des multi-masters de métadonnées incluant la détection de conflits, la personnalisation, résolution des liaisons, ...
- Gestion et résolution des conflits de données – contenus lors de toute synchronisation
- Support du changement d'unités (modification des niveaux de propriétés) et consistance des unités (échanges avec des groupes d'objets)
- Manipulation des changements et de la synchronisation de filtrage
- Rétablissement d'une multitude de scénarios d'échec, tels que le

nettoyage de périphériques corrompus, les interruptions, les échecs de réseau, etc.

- Gestion de session de synchronisation, telle que l'annulation, le reporting de progression du processus de synchronisation, etc.

Ainsi, au travers du Core Sync Runtime, le framework met à disposition un Agent de synchronisation out of the box, capable de synchroniser tout type de participant au travers de sessions de synchronisation. La notion de la diversité de connexion (de par la diversité des types de participants) est abstraite au travers de la notion de provider pour chaque source de contenu participant au processus de synchronisation. C'est également dans cette partie du framework que vous retrouverez le support de SSE (Simple Sharing Extensions) comme les flux RSS ou ATOM.



Sync Provider Framework

Comme nous l'avons parcouru, l'élément clé de cette plate-forme repose sur la notion de Provider qui définit les sources de données participant au processus de synchronisation. Nous pourrions donc, sur base du MSF, développer aisément des providers pour n'importe

quel type de données de n'importe quel support rentrant dans un processus de synchronisation. MSF fournit déjà quelques providers de base qui sont :

- Sync Service pour " ADO.NET " (**Microsoft.Synchronization.Data**) : pour la synchronisation avec ADO.NET
- Sync Service pour " File Systems " (**Microsoft.Synchronization.Files**) : pour la synchronisation de fichiers et dossiers
- Sync Service pour " SSE " (**Microsoft.Synchronization.Sse**) : pour la synchronisation des " Simple Sharing Extensions " comme les flux RSS et ATOM

Libre ensuite à tout développeur de créer ou d'utiliser d'autre Providers afin d'échanger de l'information entre terminaux et applications.

Cette notion de provider donne également naissance à une autre notion non négligeable du Sync Framework : " **la notion de participant** ". Un participant est l'endroit (terminal) où se trouve l'information ou contenu synchronisable.

Un participant peut ainsi être un webservice, un ordinateur, une clé USB, un Smartphone, un PocketPC, un TabletPC, ... Nous retrouvons ainsi dans le MSF trois types de participants sont :

- Les Participants complets (" Full Participant ") : décrit des participants où le développeur peut exécuter une application et créer des sources de données directement sur le terminal. (exemple : ordinateur portable, Smartphone, ...)
- Participant partiel (" Partial Participant ") : décrit des terminaux qui peuvent contenir des données mais ne pas exécuter d'application directement dessus. On retrouvera ici des terminaux amovibles comme des cartes SD qui peuvent contenir des données mais pas exécuter une application directement depuis le terminal.



- Participant simple (" Simple Participant ") : décrit des participants capables de fournir de l'information mais sans possibilité de stocker ou manipuler de nouvelles données. On retrouvera ici des Webservices et/ou flux RSS/ATOM.

Storage Specific Components

En plus de la capacité de créer ses propres providers de synchronisation, le MSF inclut de nombreux composants en vue de simplifier le développement de solutions hors-ligne, de synchronisation, de partage ainsi que les scénarios de roaming exploitant des protocoles ou systèmes de stockages spécifiques.

Ses composants comprennent par exemple les suivants :

- **Relational Data Providers** : ce composant comprend les services de synchronisation pour ADO.NET 2.0, la version 2.0 de SyncServices devrait être fournie avec SQL Server 2008.
- **SQL Server Compact Metadata Store** : permettant de développer des providers clients pour des terminaux ne contenant pas de solutions matures de stockage de métadonnées.
- **File and folder Sync Provider** : ce provider permet de gérer tout type de système de fichier de type Win32 (FAT, NTFS, clés USB, Disque Externe, ...).
- **ADO.NET Data Services Offline Provider**

Scénarios d'exploitation du MSF

Tentons ainsi d'explorer les scénarios pragmatiques de mise en œuvre ou exploitation du MSF. Voici quelques exemples d'exploitations proposés par Microsoft dans le cadre du MSF :

- Support hors-ligne des RIA : WebServices ou Applications Web en mode hors-ligne.
- Calendriers, contacts, listes de tâches synchronisées
- Synchronisation de bibliothèques de médias : images, vidéos, musiques, ... cette synchronisation permet par exemple de récupérer les informations de rating, ou le nombre de fois où le média a été joué sans pour autant transférer l'intégralité du fichier.
- Optimisation de la gestion du cache et ceci de manière totale ou partielle
- Partage de contenus de tout type en général, sur tout type de participant.

Développons une solution de synchronisation de données

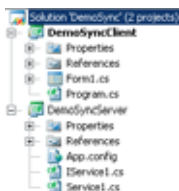
Afin de mieux comprendre son exploitation, partons de l'exemple d'une application de gestion de contacts simple, devant fonctionner en mode déconnecté. Cette application comprendra deux parties :

- Un client Winform contenant une copie en cache de la base de données sous format SDF (Microsoft Sql Server Compact)
- Une base de données centrale SQL Server de synchronisation au travers d'un service WCF.

Cet exemple sera développé à partir de Visual Studio 2008. Commençons par créer deux projets dans notre solution :

- " DemoSyncClient " de type Windows Forms Application
- " DemoSyncServer " de type WCF Service Library

Donnant le résultat suivant (Figure ci-contre).



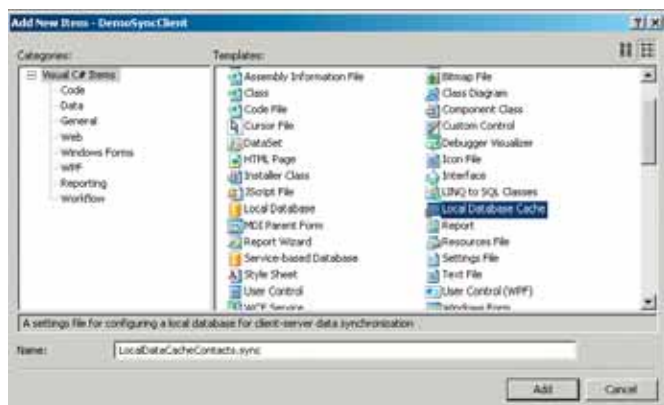
Créez une base de données SQL (ex : SQL Express) nommé " DemoSync " contenant une table " Contacts " avec les colonnes suivantes :

- "ID" (PK) de type "int" en auto-incrément
- "Nom", "Prenom", "Telephone" et "Email" en NVARCHAR

Pour cela, exécutez le code suivant :

```
USE [DemoSync]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Contacts](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Nom] [nvarchar](50) COLLATE French_CI_AS NULL,
    [Prenom] [nvarchar](50) COLLATE French_CI_AS NULL,
    [Telephone] [nvarchar](50) COLLATE French_CI_AS NULL,
    [Email] [nvarchar](50) COLLATE French_CI_AS NULL,
    CONSTRAINT [PK_Contacts] PRIMARY KEY CLUSTERED
    (
        [Id] ASC
    )WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Ajoutez un élément Local Database Cache au projet DemoSyncClient :



Note : Observez l'extension de ce nouvel Item ajouté à vos Templates d'Items de Visual Studio " sync ".

Suite à quoi, un assistant s'ouvrira afin d'y définir la base de données côté serveur (SQL Express) ainsi que la base de données client qui sera un nouveau fichier .sdf (SQL Server Compact Edition). Dans la partie " **Advanced** ", définissez le projet client (DemoSyncClient) et serveur (DemoSyncServer) afin que l'assistant génère le code dans les projets adéquats. Avant de valider, ajoutez dans les "Cached Tables" la table "Contacts" comme suit (Fig. A). En validant cette étape, un nouvel assistant se lancera afin de créer le Data-Set client de la base de données de cache (Fig. B). Enfin, supprimez les fichiers de Template IService1.cs et Service1.cs du projet DemoSyncServer qui ne nous serviront à rien. Vous devriez obtenir une solution telle que celle présentée ci-après (Fig. C)



Fig.A



Fig.B

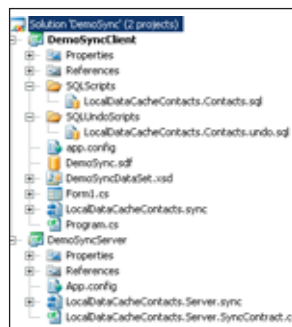


Fig.C

Vous retrouverez ainsi dans le projet client (DemoSyncClient) :

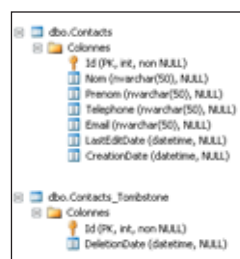
- Deux dossiers **SQLScripts** et **SQLUndoScripts** : contenant un script SQL par table permettant de préparer la table pour Sync ou de supprimer les modifications apportées pour Sync.
- Un fichier **DemoSync.sdf** qui est la version en cache de la base de données.

- Un dataset **DemoSyncDataSet.xsd** pour accéder à la base de données en cache côté client
- Un fichier **LocalDataCacheContacts.sync** permettant de reconfigurer la synchronisation

Le fichier SQL permettant la préparation de la base de données est déjà exécuté par l'assistant si vous avez laissé l'option " Execute scripts when this dialog is closed " lors de l'ajout des tables. Ce script SQL

aura ainsi automatiquement ajouté deux colonnes dans la table Contacts : " LastEditDate " et " CreationDate " permettant de comparer les données entre elles lors de toute session de synchronisation par un SyncAgent.

De plus une table " Contacts_Tombstone " est créée afin de connaître les données qui seront effacées (figure ci-contre).



Côté serveur, vous retrouverez dans le projet DemoSyncServer :

- Un fichier **LocalDataCacheContacts.Server.sync** contenant le Provider et Adapter Sync
- Un fichier **LocalDataCacheContacts.Server.SyncContract.cs** contenant le contrat et service WCF

Il vous suffit ainsi de configurer le fichier App.config du projet serveur afin d'exposer le service WCF. Pour ce faire, ouvrez le fichier App.config et supprimez dans la section "Services" le service "DemoSyncServer.Service1" créé par le template WCF Service ainsi que le behavior " DemoSyncServer.Service1Behavior " de la section " service-Behavior " par le code commenté en en-tête du fichier " LocalDataCacheContacts.Server.SyncContract.cs ".

Votre App.config devrait ainsi contenir le code suivant :

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
  </configSections>
  <connectionStrings>
    <add name="DemoSyncServer.Properties.Settings.ServerDemoSync
    ConnectionString"
        connectionString="Data Source=SEBEUHPX\SQLEXPRESS;Initial Catalog
        =DemoSync;Integrated Security=True"
        providerName="System.Data.SqlClient" />
  </connectionStrings>
  <system.web>
    <compilation debug="true" />
  </system.web>
  <system.serviceModel>
```

Pour les .Net *addicted*

Hors-Série .Net



Février / Mars 2008 - Hors-Série

www.programmez.com

Programme!
LE MAGAZINE DU DÉVELOPPEMENT

.Net **la révolution 2008** **Spécial Techdays**

WPF

**Des interfaces
riches en WPF**

Application

**Développement par
composants en .Net**

Découvrez

ASP.Net MVC :

le nouveau visage d'ASP.NET !

Mettre en oeuvre

le framework Sync

**Live : Sécurité
et communication**

Windows

**Maîtriser les
fenêtres Vista !**

.Net 3.0 / 3.5

**Démarrer avec
Windows Workflow Foundation**

Visual Studio Shell :

**Créer sa distribution
Visual Studio**

M 02104 - 10 H - F : 4,95 € - RD

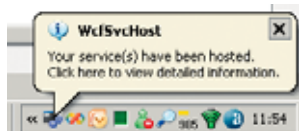


Printed in France - Imprimé en France - BELGIQUE 5,45 € - SUISSE 10 FS - LUXEMBOURG 5,45 € - Canada 7,45 \$ CAN - DOM Surf 5,90 € - TOM 780 XPF - MAROC 50 DH

Parution le 9 février 2008

```
<services>
  <service name="DemoSyncServer.LocalDataCacheContactsSync
Service" behaviorConfiguration="DemoSyncServer.LocalDataCache
ContactsSyncServiceBehavior">
    <host>
      <baseAddresses>
        <add baseAddress="http://localhost:8080/LocalDataCache
ContactsSyncService/" />
      </baseAddresses>
    </host>
    <endpoint address="" binding="wsHttpBinding" contract="Demo
SyncServer.LocalDataCacheContactsSyncContract" />
    <endpoint address="mex" binding="mexHttpBinding" contract=
"IMetadataExchange" />
  </service>
</services>
<behaviors>
  <serviceBehaviors>
    <behavior name="DemoSyncServer.LocalDataCacheContactsSync
ServiceBehavior">
      <serviceMetadata httpGetEnabled="True" />
      <serviceDebug includeExceptionDetailInFaults="False" />
    </behavior>
  </serviceBehaviors>
</behaviors>
</system.serviceModel>
</configuration>
```

Votre service sera ainsi exposé sur l'adresse <http://localhost:8080/LocalDataCacheContactsSyncService/> en utilisant le binding " wsHttp-



Binding " Référez ce service dans le projet client et compilez votre solution et cliquez-droit sur le projet client pour sélectionner " Add Service Reference ".

Entrez l'URI de votre service ainsi que le nom de la référence et validez afin de créer le délégataire client de votre service.

La solution est maintenant prête pour MSF. Créez alors une interface de manipulation des données et de contrôle de la synchronisation.

Pour cela, profitez des avantages du DataSet au travers de la fenêtre d'outils " Data Sources " à partir de laquelle vous pouvez glisser-déplacer les champs du DataSet désirés dans le Form comme sur la figure D. En exécutant le projet client vous pourrez ainsi créer, modifier ou supprimer des contacts.

En consultant les données de votre table sur votre serveur SQL vous verrez qu'aucune donnée n'est pourtant présente. Cela s'explique par le fait que toutes les données manipulées côté client sont stockées sur la base de données SQL Compact (.sdf) et non sur le serveur. Il vous faudra maintenant configurer la synchronisation de la base de données client (sdf) avec notre serveur SQL au travers du service WCF. Ajoutez un bouton dans notre Form et exécutez le code suivant lors du clic :

```
private void btSynchronise_Click(object sender, EventArgs e)
{
  // Agent sync
  LocalDataCacheContactsSyncAgent syncAgent =
```

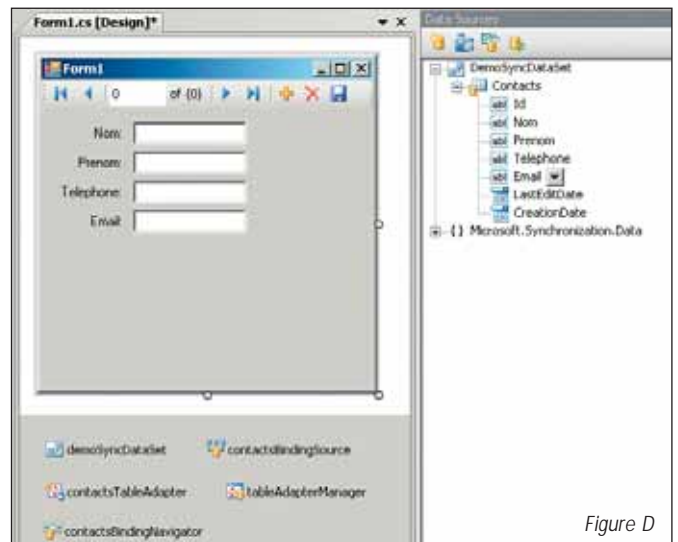


Figure D

```
new LocalDataCacheContactsSyncAgent(
    new ServiceSyncContacts.LocalDataCacheContactsSync
ContractClient());

// Synchronisation des contacts bi-directionel
syncAgent.Contacts.SyncDirection =
    Microsoft.Synchronization.Data.SyncDirection.Bidirectional;

// Synchronisation
Microsoft.Synchronization.Data.SyncStatistics syncStats =
    syncAgent.Synchronize();

// Mise à jour du DataSet
this.contactsTableAdapter.Fill(this.demoSyncDataSet.Contacts);

// Affichage des résultats
MessageBox.Show(string.Format("Start: {2}\nCompleted: {3}\nDL
Changes applied : {0} - failed : {1}\nUP Changes applied : {6} - failed :
{7}\nTotal changed downloaded : {4} - uploaded : {5}",
    syncStats.DownloadChangesApplied,
    syncStats.DownloadChangesFailed,
    syncStats.SyncStartTime.ToString(),
    syncStats.SyncCompleteTime.ToString(),
    syncStats.TotalChangesDownloaded,
    syncStats.TotalChangesUploaded,
    syncStats.UploadChangesApplied,
    syncStats.UploadChangesFailed));
}
```

Votre application est désormais fonctionnelle. Elle permet de manipuler des contacts en se basant sur un cache du serveur SQL dans le fichier local " DemoSync.sdf " et permet la synchronisation de cette base de données locale avec le serveur SQL au travers d'un service WCF.

■ **Grégory Renard**
CTO Wygwan
Microsoft Regional Director
Microsoft MVP

■ **Sébastien Warin**
Développeur .NET - Wygwan
Microsoft MSP

Volta : un projet multifonction

Projet en vogue en ce moment sur les Labs Microsoft, Volta a tout de suite été comparé à GWT (Google Web Toolkit), voire même pour certains, au nouveau Java Killer made in Microsoft. Mais qu'en est-il réellement ? Sur quels fondements repose Volta et à quoi sert ce nouveau projet de recherche ?

C'est au cours de l'été que j'ai entendu parler de Volta pour la première fois lors d'une visite sur le site « Channel 9 » (<http://channel9.msdn.com>) en tombant sur deux vidéos intitulées « Volta - Wrapping the Cloud with .NET » mettant en scène Erik Meijer, membre de l'équipe Microsoft Research. J'étais impatient de voir la première CTP de ce projet, encore un peu flou, mettant en scène la notion de « Tiers Splitting » et il a fallu patienter jusqu'au mois de Décembre pour voir apparaître la première release publique.

Volta est aujourd'hui un projet de recherche et non un produit. Nous savons qu'une cellule a été montée au sein de l'équipe de Redmond, mais nous n'avons, à ce jour, aucune information concernant la finalisation produit de cette technologie, ni aucun agenda de développement (roadmap). Ce Labs est actuellement en ligne de façon à faire découvrir à la communauté les fondements de Volta et obtenir des retours utilisateurs quant au potentiel d'une telle technologie. Après avoir lu beaucoup de choses sur Volta, j'ai finalement creusé plus en profondeur ce projet afin de comprendre réellement ce qui se cachait derrière.

Volta : un compilateur d'IL !

Le premier fondement de Volta réside dans la capacité à analyser de l'IL (Intermediate Language) pour le compiler sous une autre forme. A la différence de GWT qui, dans les grandes lignes, analyse du Java pour produire du JavaScript, Volta se veut plus généraliste au sein de la plate-forme Microsoft. Effectivement limité aujourd'hui aux WebForms et WinForms dans la première CTP, nous pouvons déjà imaginer que d'autres types de projets seront à terme intégrés au sein de Volta. Reposant sur de l'analyse IL, il est fort envisageable de voir apparaître des projets de type SilverLight et/ou WPF.

Le but de cette analyse est de pouvoir produire un code différent en sortie. Par exemple, l'analyse de l'IL d'un projet Volta Application générera du code JavaScript pour exécuter notre application au sein d'un navigateur. Mais Volta est aussi capable d'analyser de l'IL pour produire de l'IL, ce que nous retrouvons au sein des projets Volta Windows Forms. Une utilité de ce procédé réside dans l'abstraction. Par exemple, Volta est livré avec une encapsulation de Virtual Earth vous permettant d'utiliser l'API en C#/VB.Net sans pour autant connaître les subtilités de l'API JavaScript. C'est l'analyse de l'IL produit qui générera les différents appels JavaScript nécessaires à l'utilisation de Virtual Earth au sein de votre projet.

Volta : Tiers Splitting

Terme étrange au premier abord, le « Tiers Splitting » est une série d'outils intégrée à Visual Studio permettant le Refactoring du code. Cette notion va nous permettre de décider au dernier moment dans quel contexte s'exécutera notre code en ajoutant simplement quelques attributs sur nos classes et/ou méthodes. C'est au travers de ces attributs que nous déciderons, une fois le code implémenté, si notre méthode

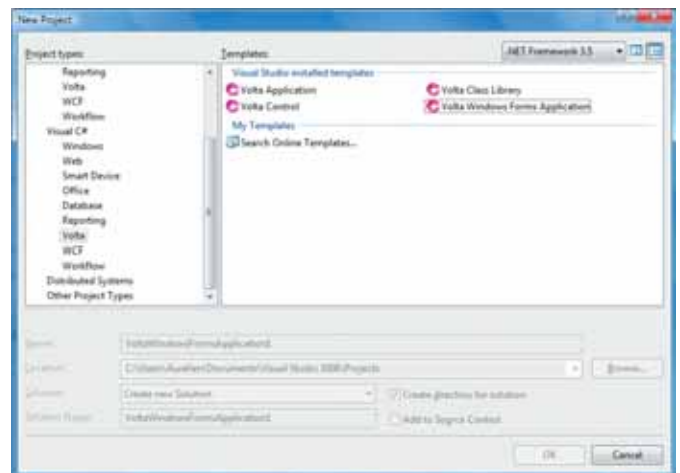
s'exécutera côté Client ou côté Serveur. C'est ensuite au travers de l'analyse de l'IL que Volta compilera le code nécessaire à une exécution « Full Client » ou « Client / Serveur », et ce, quel que soit le projet de départ.

Pour cela, Volta est fourni avec un WebServer intégré qui se chargera de l'exécution du code marqué par l'attribut [RunAt(« Server »)]. Dans le cadre d'un projet type WinForm, l'IL est recompilé de façon à appeler le WebServer pour l'exécution de notre code.

Cette méthodologie par attributs est assez sympathique car elle ne nécessite aucune modification de votre code pour gérer la communication entre le Client et le Serveur. On peut déjà imaginer la puissance d'un tel outil dans la rapidité de mise en place de ce type de projets.

Exemple Volta Windows Forms Application

Pour cela, il vous faudra posséder Visual Studio 2008 (RTM ou Bêta 2) et aller sur <http://labs.microsoft.com/volta/download> afin de récupérer la CTP. Une fois Volta installé, 4 nouveaux templates de projet seront présents au sein de Visual Studio.



Nous allons donc ici créer un projet de type « Volta Windows Forms Application » nommé *VoltaWindowsFormsSample*. Afin de bien comprendre la notion de Tiers Splitting, nous allons ajouter une classe dans notre projet, nommée *ProcessService* contenant le code suivant :

```
namespace VoltaWindowsFormsSample
{
    public class PInfo
    {
        public string Name { get; set; }
        public int Id { get; set; }
    }
}
```

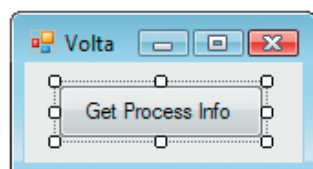
```

}

public class ProcessService
{
    public static PInfo GetProcessInfo()
    {
        return new PInfo {
            Name = System.Diagnostics.Process.GetCurrentProcess().ProcessName,
            Id = System.Diagnostics.Process.GetCurrentProcess().Id
        };
    }
}

```

Comme vous pouvez le constater, rien de bien complexe, nous regardons juste le process courant afin de récupérer son nom et son identifiant.



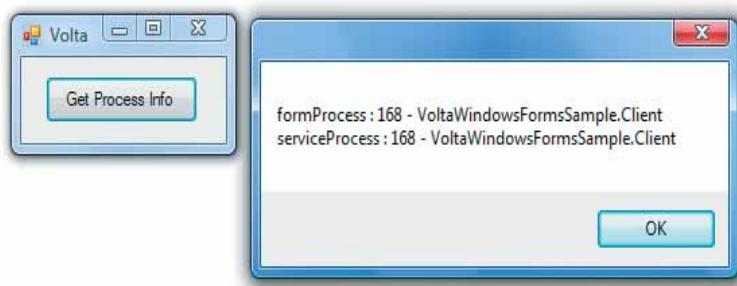
Ensuite, dans notre formulaire de base, nous allons ajouter un bouton nommé *btnProcessInfo* pour arriver au résultat ci-contre. Le clic sur ce bouton exécutera le code suivant :

```

private void btnProcessInfo_Click(object sender, EventArgs e)
{
    PInfo formProcess = new PInfo()
    {
        Name = System.Diagnostics.Process.GetCurrentProcess().ProcessName,
        Id = System.Diagnostics.Process.GetCurrentProcess().Id
    };
    PInfo serviceProcess = ProcessService.GetProcessInfo();
    MessageBox.Show(
        string.Format("formProcess : {0} - {1} \n serviceProcess : {2} - {3}",
            formProcess.Id,
            formProcess.Name,
            serviceProcess.Id,
            serviceProcess.Name
        )
    );
}

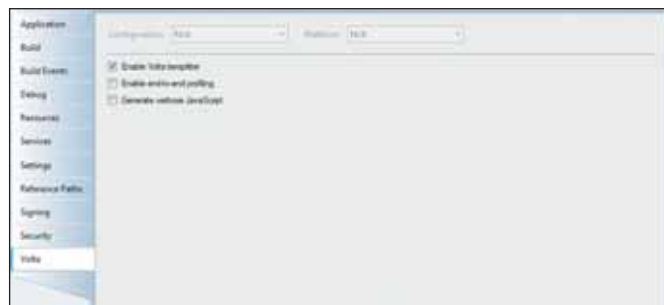
```

Toujours rien de bien compliqué derrière ce code, nous nous contentons juste de récupérer le process courant afin de le comparer au process récupéré via notre classe. Une fois le code implémenté, nous lançons le projet en mode debug pour obtenir le résultat suivant :

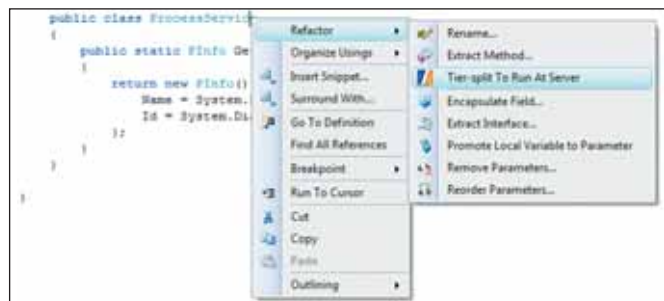


Rien de surprenant dans cette exécution, nous sommes bien en mode « Full Client ».

Nous allons maintenant utiliser le Tiers Splitting de Volta pour déporter l'exécution de la méthode *GetProcessInfo* sur le serveur. Pour cela, nous devons dans un premier temps autoriser le Tiers Splitting au sein de notre application au travers des propriétés de notre projet.



Nous allons ensuite utiliser les outils intégrés à Visual Studio pour exécuter notre méthode sur le serveur. Cela aura pour effet d'ajouter l'attribut *[RunAt(« Server »)]* sur notre classe *ProcessService*.



Nous relançons le projet en mode debug, et nous remarquons de suite qu'un WebServer Volta démarre en parallèle de notre WinForm.

Un double clic sur l'icône du WebServer, Volta ouvrira une fenêtre nous permettant de voir les requêtes émises vers ce dernier. Nous ouvrons donc le WebServer pour cocher la case de log des requêtes, nous cliquons sur notre bouton et nous remarquons de suite que la réponse n'est pas aussi instantanée que sur le premier exemple.

La capture précédente montre bien que notre application a été recompilée afin de modifier le code IL pour faire un appel de la page *__TierSplit.aspx* sur Volta WebServer afin d'exécuter le code marqué par l'attribut *[RunAt(« Server »)]*. Cette exécution délocalisée peut se vérifier en regardant les process. L'un est exécuté au sein de notre WinForm *VoltaWindowsFormsSample.Client*, l'autre au sein de *VoltaWebServer*. Cet exemple bénin nous montre toutefois les capacités de Volta quand à la recompilation de code IL et du « Tiers Splitting ».

Conclusion

Comme tout projet de recherche, Volta possède aussi des points faibles, notamment dans les optimisations de performances. Par exemple, le code JavaScript généré par une « Volta Application » est loin d'être optimal pour le moment et impacte directement le temps de chargement. Cependant, un des derniers billets du blog de l'équipe Volta laisse présager un remplacement de la génération JavaScript par du SilverLight 2.0 !

■ Aurélien Verla

Online Strategies Team Manager - Wygwam - <http://www.wygwam.com>

L'INFORMATION du DÉCIDEUR

Choisir, déployer, exploiter les logiciels

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

Egalement au sommaire du numéro :

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



L'actualité au quotidien :

- Sécurité • Projets
et développement
- Administration
- Progiciels

Les Cas Clients

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



N°1 - Février/Mars 2008
Chez votre marchand de journaux
Abonnez-vous (coupon ci-dessous)

www.solutions-logiciels.com

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

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

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

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

☐ M. ☐ Mme ☐ Mlle Société

Titre : Fonction : ☐ Directeur informatique ☐ Responsable informatique ☐ Autre

NOM Prénom

N° rue

Complément

Code postal : Ville

Adresse mail

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



La base de données : au cœur des applications

Plus que jamais, le SGBD est au cœur des développements, des applications, ou plus précisément, de la donnée. Malgré les progrès dans l'accès à cette dernière, le développeur reste souvent distant, voire réfractaire. La communication entre développeur et DBA ne s'améliore pas beaucoup. Or, ils ont tout intérêt à travailler ensemble pour optimiser, simplifier les accès. Sans connaissance de la donnée et de la base, le développeur sera tôt ou tard confronté à un problème : comment exposer, consommer, gérer la donnée. Dans les projets importants, le développeur ne peut se passer du DBA et vice versa. Pourquoi ? Le DBA apporte la connaissance de la base, sa structure, son architecture. Le développeur est le seul à pouvoir coder, optimiser les requêtes et procédures stockées, le seul à être connecté physiquement au code, à l'applicatif. Bref, les rôles sont clairement définis. Cependant, l'idée de voir un développeur de données n'est plus guère d'actualité contrairement à ce que l'on aurait pu penser il y a encore un an. Cependant, le développeur doit acquérir des compétences SGBD et si possible sur plusieurs bases. La maîtrise du SQL, des outils de mapping, sont deux autres prérequis. Sans cette connaissance, difficile d'optimiser les accès.

Optimiser sa base

Car le credo reste toujours le même : on peut rajouter de la mémoire, des serveurs, des bases, si le requêtage, les procédures stockées sont mal codés, vous ne gagnerez rien sur les performances intrinsèques entre l'application et le serveur de données. Plus que jamais, la maîtrise de ces techniques s'impose. La complexité des bases, des tables, des applications, l'accroissement des données requièrent une rigueur toujours plus grande. C'est là aussi le rôle de l'architecte qui doit avoir une vision globale, y compris la base. Aujourd'hui, les nouvelles bases de données impactent relativement peu les fondamentaux pour le développeur. Les langages SQL internes évoluent mais rien de cassant avec



l'existant. Seul Microsoft a innové sur la question de l'accès en proposant un requêtage direct à partir du langage de programmation avec Linq. Ou alors il faut passer en pur objet avec un SGBD de type InterSystems Caché qui évite de passer par une couche supplémentaire. Car on bute toujours sur deux mondes : d'un côté le monde généralement objet et le monde relationnel. Les deux ne communiquent pas directement, d'où l'usage de requêtes, de frameworks de mapping pour assurer l'accès. Comme on le sait tous, plus on empile des couches, plus on complexifie, plus on fragilise. Avec la multiplication des architectures de services et autres applications composites et Software + Service, il faudra bien suivre les chemins d'accès aux données... Et on sera dans un monde de plus en plus hétérogène, car deux services peuvent avoir une source de données différente... Bref, le développeur travaillera en abstraction, aux services et aux frameworks de faire le reste.

Priorité aux nouveautés

Pour ce dossier, nous avons opté pour deux approches. La première plutôt axée sur : com-

ment choisir sa base, quels critères, fonctions faut-il considérer. Et aujourd'hui ce choix est difficile car les solutions, qu'elles soient open source ou commerciales, se rapprochent de plus en plus. On ne peut plus trancher uniquement sur les fonctions fondamentales. Les réelles différences se font sur les volumétries très importantes, éventuellement les outils de développement disponibles, sur les standards ou encore par rapport à son environnement serveur, même si la virtualisation brouille la donne et tend à fournir un socle serveur de plus en plus universel sur lequel on peut mettre n'importe quelle base. Le second axe concerne les SGBD eux-mêmes. Nous avons donc privilégié les environnements et trois d'entre eux tout particulièrement : Oracle 11g avec les évolutions sur le PL/SQL, SQL Server 2008 et le retour en grâce du DBA après la montée en puissance du développeur (dans SQL Server 2005) et enfin, nous aborderons une base de données historique : Ingres, SGBD open source, encore trop méconnu par les développeurs et qui concurrence directement MySQL, Firebird, et autres PostgreSQL.

■ François Tonic

SGBD 2008 : évolution, continuité, rachat

Le millésime 2008 a débuté dès septembre dernier avec Oracle 11g. Et durant toute cette année, le marché va accueillir deux nouvelles versions importantes : SQL Server 2008 (mai ou juin prochain), MySQL 6.0 (fin de l'année, malgré le rachat par Sun), sans compter des nouveautés chez Sybase, Firebird, Ingres, Pervasive, PostgreSQL, etc. Devant un tel choix, on pourrait résumer le dilemme ainsi : " je veux faire quoi avec les données ? "...

Depuis 2007, les éditeurs se positionnent clairement sur les applications web et tout spécialement sur le " web 2 ", les services en ligne, le Software + Service (S+S). MySQL se veut le partenaire idéal de l'entreprise dite 2.0... Et il est vrai que les applications monolithiques explosent pour s'exposer sur le navigateur en interne ou en externe, via des services dans la SOA ou le S+S. Derrière, il y a les données, les informations, donc la base de données. Sur la mobilité, le marché commence lui aussi à bouger sérieusement. Là, MySQL ne propose rien et il n'y a pas de prévisions sur l'agenda (même si techniquement rien ne s'y oppose, Nokia l'a fait en modifiant une base MySQL), à l'inverse d'un Sybase ou Microsoft pour qui, la base mobile est un axe important. D'autre part, si depuis 2 ans, les éditeurs mettaient systématiquement en avant XML (stockage, gestion, transformation), aujourd'hui, XML n'est plus l'argument prioritaire. Pour preuve, MySQL qui commence à le supporter se montre particulièrement prudent. Aux utilisateurs de dire s'il faut ou non continuer dans cette voie.

Vers une base agnostique ?

InterSystems, avec sa base Caché, reste très Java dans son approche, orientée Web 2 avec le framework Zen, SQL Server 2008 reste .Net. Pour les autres, tout va dépendre de leurs priorités. MySQL 6.0 conserve une approche neutre en supportant PHP, Java, .Net, tout comme DB2 Viper. Avec la composition de services, de S+S, de la SOA, des applications composites, il s'agira d'utiliser des services ayant des sources de données différentes. Il faudra alors être capable d'intégrer tout cela. Mais chaque base garde ses spécifications. Pas question d'accueillir une base d'un autre SGBD sans faire de migration. Le rôle des outils de développement va aller croissant. Il s'agit aussi pour les édi-

teurs de ne pas casser avec les versions précédentes. Ainsi si entre SQL Server 2000 et 2005, il y avait une vraie rupture et une migration pas toujours facile, entre 2005 et 2008, pas de rupture. " Continuité par rapport à SQL Server 2005. On renforce les usages de la version 2005 " précise Lionel Billon (Microsoft France - Chef de produit SQL Server).

Les fonctions majeures de 2008

Il n'y a pas de réelles tendances nouvelles dans les SGBD, plutôt des évolutions de fonctions existantes. C'est le cas dans le partitionnement qui s'impose partout. Il n'impacte pas le développeur car sa gestion est avant tout liée au DBA. Pour l'application, le partitionnement est transparent, tout comme le load balancing, le clustering, la réplication. Pour l'utilisateur final, il peut y avoir un ralentissement de fonctionnement (par exemple en cas de crash d'une base le temps qu'un autre serveur prenne le relais).

Build à SGBD : nouveau défi ?

Les outils constituent de nouveau un axe de développement pour les éditeurs de SGBD. MySQL propose, outre le plug-in pour Visual Studio, un environnement complet : MySQL Workbench. Il s'agit d'un éditeur visuel pour designer sa base MySQL. Disponible en version gratuite et commerciale. Il y a des différences fonctionnelles sur les modèles, la synchronisation. IBM propose Data Studio qui sert à designer, développer, déployer et gérer les données, les bases et applications orientées données. Il s'intègre dans la notion de cycle de vie des données. Ce cycle reste encore discret mais la gestion de la donnée devient une réelle problématique, surtout quand on utilise plusieurs sources et une volumétrie importante. Cela suit quelque peu la philosophie de Microsoft avec Team Edition for Database Professionals. Ce type d'outil est la plupart du temps lié à une base précise. Nous reviendrons sur cet aspect spécifique dans le prochain numéro de Programmez !.

■ J.V.

SGBD en mémoire, SGBD mémoire, serveur mémoire

Une des solutions pour accélérer les accès, les bases est l'emploi de SGBD en mémoire ou encore des moteurs en mémoire. Dans le cas de MySQL, on peut mettre en œuvre SolidDB for MySQL. Entièrement optimisé pour le multicore et le multithreading, il permet de booster sa base MySQL classique. Dans une future version, il sera possible de mettre en œuvre la notion de base de données en mémoire. Les SGBD en mémoire sont particulièrement optimisés en utilisant la mémoire vive, le cache, le processeur multicœur et le 64-bit. Avant tout destiné aux bases critiques ou à forte contrainte de performances, il s'agit d'un marché encore marginal mais qui pourrait croître dès 2008. IBM a investi dessus en rachetant l'éditeur solid qui édite SolidDB. Oracle dispose de sa propre solution : TimesTen. Dans le monde embarqué, ce type de SGBD n'est pas rare, comme la solution eXtremDB. Par définition, la base en mémoire tourne dans la mémoire vive principale. Les SGBD Java fonctionnent ainsi mais les performances varient énormément.

Cette solution reste onéreuse tout comme les serveurs de mémoire en remplacement des disques durs serveurs classiques. Ce type de stockage est encore trop récent pour pouvoir l'envisager sur une base critique.

Comment choisir sa base de données ?

Le choix d'un SGBD ne s'effectue pas à la légère. Performances, capacité de montée en charge, coûts de fonctionnement ou facilité de développement sont quelques-uns des critères à étudier pour trouver la base adaptée aux besoins de l'entreprise.

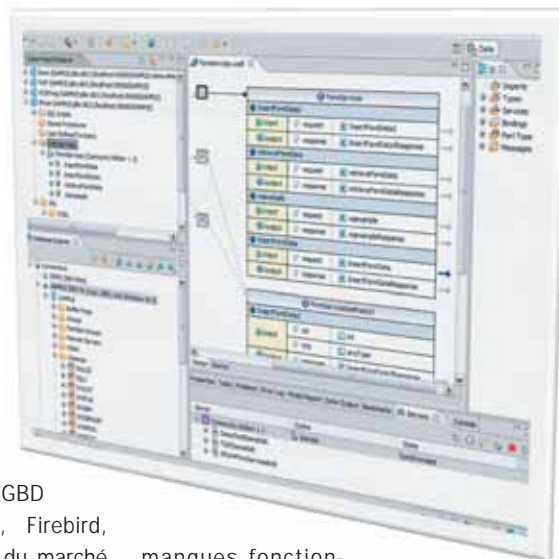
Oracle, DB2, SQL Server, MySQL, Firebird, Ingres, Sybase... Difficile de s'y retrouver lorsque l'on cherche une base de données adaptée à ses moyens et besoins. Le choix est d'autant plus ardu qu'il dépend à la fois de critères techniques et subjectifs. Côté technique, les éditeurs proposent des solutions dont la richesse fonctionnelle varie fortement. Si les entreprises, dotées de DBA (administrateurs de bases de données) et d'équipes de développement, peuvent réaliser des tests et études approfondies, le choix du SGBD " idéal " est plus complexe pour une PME qui ne dispose pas ou peu d'administrateur ni des compétences nécessaires. Grands comptes ou PME, tous sont cependant égaux vis à vis des politiques d'entreprises actuelles qui visent, la majorité du temps, à réduire les budgets de fonctionnement dédiés aux systèmes d'informations. Dans les grands comptes, rationalisation des dépenses oblige, les responsables informatiques ne sont plus, en outre, seuls à décider de l'utilisation de leurs budgets. De nouveaux paramètres rentrent également en compte comme la politique de l'entreprise vis à vis des fournisseurs, les budgets disponibles, et, côté technique, l'existant, c'est-à-dire les SGBD, les systèmes d'exploitation et l'infra-

structure matérielle en service, ou encore l'évolution technique et technologique de la base de données (par exemple : faut-il migrer sur la nouvelle version ou changer de base). Comment choisir objectivement entre un SGBD Open Source (MySQL, Ingres, Firebird, PostgreSQL...) et les historiques du marché (Oracle, Sybase, Microsoft, etc.) ? Le nom et la réputation suffisent-ils à exonérer le DSI de toute remontrance en cas de problème lors de l'implémentation de la nouvelle base ?

Bases gratuites, en Open source ou traditionnelles ?

Pourquoi payer lorsqu'une base de données est disponible gratuitement (tout du moins au niveau licence) ? C'est le cas des bases Open Source comme MySQL (excepté pour les licences entreprises et commerciales), Cloudscape (plus orientée base embarquée, enfouie), PostgreSQL ou un Firebird. Chez les éditeurs commerciaux, les versions gratuites existent aussi : SQL Server Express, Oracle Database Express, Sybase, DB2 C-Express. Ces bases gratuites sont a priori séduisantes, mais possèdent souvent des limitations et des

manques fonctionnels. Tout dépend des besoins de l'entreprise. Les bases Open Source conviennent par exemple à des applications peu gourmandes en ressources et ne nécessitant pas de mise en œuvre en cluster (même si, bien entendu, ces bases open source sont largement utilisées pour les données et applications critiques). Un SQL Server Express n'est pas conçu pour les applications fortement transactionnelles. Même chose en ce qui concerne les capacités de prise en charge d'un nombre important d'utilisateurs simultanés, les fonctions de redondance ou d'équilibrage de charge, le nombre de processeurs, la consommation de mémoire vive. Ces dernières ne seront peut-être pas disponibles sur une de ces bases gratuites tout comme les mécanismes d'index bitmap adaptés aux recherches sur de grandes volumétries de données ou la partition de tables sur plusieurs disques. Autre problématique, pour les applications critiques et devant toujours être disponibles : la base doit être administrée à chaud, sans jamais s'arrêter. Peu de SGBD libres le permettent. Ces fonctions avancées sont traditionnellement l'apanage des bases traditionnelles comme Oracle, DB2, Informix ou SQL Server, implantées depuis longtemps sur le marché. Il est indispensable de bien étudier ses besoins de façon à éviter les mauvaises surprises liées aux fonctions avancées qui peuvent aller jusqu'aux mécanismes décisionnels, au stockage de données XML ou de coordonnées géographiques nécessaires aux applications de géomarketing.



Quelle base pour quels usages ?

Pour choisir son SGBD, une multitude de critères objectifs et subjectifs entrent en ligne de compte. Il est donc très difficile d'avoir une réponse toute faite à la question " quelle base pour quels usages ? ". En revanche, il est des points techniques sur lesquels il faut se pencher dès les premiers jours du projet :

- la criticité de l'application et les usages qui vont en être faits : transactionnel, décisionnel, consultation...
- les fonctionnalités du SGBD et leurs limitations, notamment en terme de partitionnement, de capacité de stockage, de nombre de connexions simultanées, de nombre de transactions traitées par seconde, de traitement de fortes volumétries...
- les ressources matérielles (disques, mémoire, CPUs, support de la virtualisation...)
- les compétences existantes en terme de développement et d'administration, et au besoin le prix de la formation
- les langages de développement supportés

Boostez vos Applications !



Embarquez le moteur de développement le plus rapide et faites décoller vos applications !

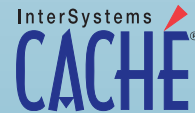
InterSystems Caché® offre à tous les développeurs rapidité, scalabilité et portabilité maximale. Caché® est le moteur de base de données et de développement le plus rapide que vous puissiez intégrer à vos applications et qui vous offre simultanément les avantages des environnements Objet et Relationnel, tout en diminuant les coûts d'administration et de matériel.

Développez sans mapping! Grâce à son architecture novatrice, Caché épargne aux programmeurs Objet (Java, .NET, etc) des efforts laborieux en éliminant le mapping objet-relationnel.

Développez où vous voulez! Une application Caché développée sous un OS tournera sous un autre OS sans effort de portabilité (Caché est disponible sous Unix, Linux, Windows, Mac OS X, et OpenVMS.)

Développez vraiment rapidement! Caché intègre un environnement de développement puissant ainsi qu'un framework de type Ajax. Caché peut être utilisé comme serveur d'application et supporte également en standard le T/SQL ou MultiValue.

Caché est déjà déployé sur plus de 100,000 systèmes, supportant jusqu'à plus de 50,000 utilisateurs. Embarquez nos innovations et enrichissez vos applications !



Make Applications More Valuable

Téléchargez une version complète de Caché gratuitement ou demandez votre CD sur InterSystems.fr/embarquez

Les coûts cachés : éternel serpent de mer !

Même Open Source, une base n'est jamais gratuite. Si MySQL, PostgreSQL sont disponibles gratuitement, le support et la maintenance sont payants et dans certains cas, il faut acquérir des licences commerciales pour une utilisation commerciale ou en entreprise. Bien entendu, cela dépend aussi des compétences disponibles en interne, notamment de celles des développeurs. Le problème de support, de l'assistance est identique pour les bases commerciales.

La mise en œuvre des fonctions avancées demandera, souvent, le recours à une société de service, qui assurera le transfert de compétences vers les équipes internes, et à des formations techniques du personnel, deux options dont les coûts viennent s'ajouter aux frais de licence de la base. Il ne faut pas oublier non plus la partie matérielle, l'administration globale de la base, les coûts de migration s'il y a lieu, l'impact au niveau applicatif et architecture. Et aujourd'hui, avec la génération du multicore, il faut bien comprendre la facturation au processeur (par processeur physique ou par cœur, ce qui peut faire varier la note finale). Et avec la génération de la virtualisation, il est aussi nécessaire de bien lire les licences d'utilisation et ce qu'on est autorisé ou non à faire. Par exemple, peut-on virtualiser son SGBD à volonté ?

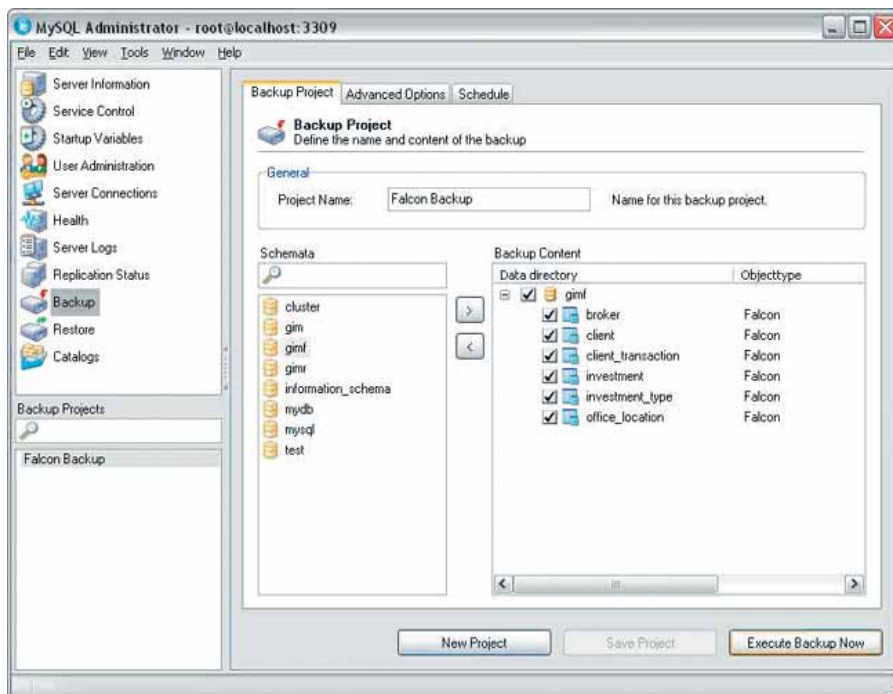
Mesurer les performances de sa future base

Une fois le cahier des charges réalisé (souvent après un audit interne ou réalisé par des consultants), reste à s'assurer des capacités techniques de la base. La plupart des éditeurs publient les performances de leurs produits au travers de tests TPC-C. Réalisés par des organismes indépendants, ces tests sont fiables, mais présentés de manière avantageuse par les éditeurs, toujours prompts à mettre en avant les points forts de leurs SGBD. Il s'agit d'un premier niveau d'information. C'est à vous, entreprise et développeur, de tester, re-tester les fonctions, les capacités du SGBD. Rien ne vaut une mise en œuvre dans un projet pilote ou une maquette qui permettra également de vérifier la bonne adéquation entre les besoins et l'outil, et la capacité de montée en charge future de la base.

Le SGBD d'un site de commerce électronique devra, par exemple, gérer plusieurs milliers d'accès simultanés. Le maquettage est une



Oracle 11g



Module d'administration de MySQL.

démarche normale des grands comptes, mais pas toujours évidente dans une petite structure. Là encore, le recours à une société de services sera une source de coûts supplémentaires mais indispensable pour bien choisir.

Et l'administration des bases ?

Exploiter sa base au quotidien nécessite des outils d'administration capables de détecter entre autres, les goulets d'étranglements, les baisses de performances, les entrées-sorties ou le manque d'espace disque, d'assurer le partitionnement, de créer des politiques de fonc-

tionnement (réplication, etc.) etc.. Les SGBD arrivent avec leurs propres consoles d'administration. C'est un point à évaluer : la console standard est-elle suffisante ? Ou faut-il opter pour un outil tiers ? Autre aspect incontournable : la mise en œuvre de fonctions d'audit et de sécurisation de la base, qui permettent d'être alertées en temps réel des événements de sécurité critiques tels que les détections d'intrusion, les tentatives de vol de données ou de fraude. Cette technologie n'est pas mise en œuvre par toutes les bases (en particulier les bases Open Source) et nécessitera de se tourner vers les versions les plus récentes des

Critères et tests de sa future base

Choisir une base n'est jamais évident. L'installation peut être un des facteurs de choix. Car s'il faut une journée ou deux journées pour déployer entièrement le SGBD, ce n'est pas la même chose. MySQL sort du lot dans le sens où la base est présente dans les distributions GNU Linux, dans les offres LAMP et WAMP, et de nombreux éditeurs l'embarquent. Mais il arrive souvent que le client, l'entreprise impose son choix.

SQL, est-il un critère pertinent ?

Non, plus maintenant. Si l'accès aux données est un élément important, SQL est aujourd'hui présent partout et, selon l'éditeur, il sera d'un support plus ou moins avancé. Ensuite, il y a les dérivés spécifiques à chaque éditeur comme c'est le cas chez Oracle et Microsoft. Le PL/SQL d'Oracle est particulièrement riche, côté Microsoft, le support .Net peut être un plus dans un contexte purement Windows et l'arrivée de Linq peut être un avantage. Ensuite, le développeur peut regarder le niveau des pilotes d'accès aux données (ODBC, JDBC) ou encore les possibilités de mapping de données.

La volumétrie : un faux argument !

Toutes les bases gèrent sans problèmes 100, 300 ou 500 Go de données, que ce soit dans une base, en partitionnement, en cluster. Ce n'est plus un critère décisif de choix comme cela pouvait l'être il y a quelques années. Cependant, pour les très gros volumes, IBM DB2 reste une référence, les environnements open source seront en retrait. À moins de 500 Go tout est gérable.

Dans les grands volumes, les fonctions de partitionnement sont souvent vitales. Pour le développeur, cela n'a pas un grand impact car c'est au niveau DBA que cela se joue réellement. Les principales bases gèrent et proposent plusieurs types de partitionnement. Au-delà de la volumétrie, il y a la question de la disponibilité de la base, des données, la

reprise d'activité après un crash matériel ou logiciel, la montée en charge, le stockage. Mais là, les principales bases offrent du cluster, de la haute disponibilité, de la montée en charge. Un des critères est de savoir si ces fonctions sont livrées en standard ou proposées en option.

Si j'ai besoin de plusieurs moteurs de stockage

Sur ce point, MySQL sera votre choix. C'est le seul à proposer une architecture modulaire avec la possibilité de changer le moteur. Mais là se pose la question de la pérennité de ces moteurs. Si vous choisissez un moteur et que celui-ci n'est plus supporté, que faire ? C'est en partie pour cette raison que MySQL développe pour MySQL 6.0 son propre moteur : Falcon. Pour un grand compte, mais même pour une petite entreprise, la pérennité est un argument important. Et trop de choix peut rebuter une entreprise car elle n'aide pas forcément à choisir. Par contre, les standards constituent un autre argument de poids, à l'heure de l'interopérabilité, de l'intégration, des échanges.

Décisionnel : un marché encore commercial

Sur le décisionnel, les bases commerciales gardent un petit avantage, même si MySQL et PostgreSQL améliorent les choses. SQL Server représente un choix intéressant par son approche tout en un, alors que chez Oracle, il faudra passer par des options.

Le système comme déterminant

Si vous travaillez en dehors de Windows, vous pouvez tout de suite, supprimer de la liste

SQL Server, fonctionnant uniquement sous Windows. Sur le marché Linux, Oracle et IBM font un véritable forcing. En entreprise, la question se pose de prendre des blades ou serveurs sous Linux, le coût du matériel étant bien moins élevé que de prendre du mainframe. MySQL est bien entendu un acteur important sur les principaux systèmes. Dans certains cas, on peut utiliser une version gratuite des bases commerciales, la différence se fera sur les capacités et certaines fonctions avancées, sans pour autant simplifier la base. Si la base est difficile à installer, la version express ne modifiera pas les choses. Par contre, pour de petits projets, une maquette, une base express peut se justifier.

La question du support

Un élément qui conserve toute son importance : le support et l'assistance. La qualité varie beaucoup d'un éditeur à un autre. La communauté peut être d'un bon recours en cas de problème fonctionnel. C'est une des forces des bases open source et même de Microsoft. Avec le rachat de MySQL par Sun, l'éditeur a annoncé sa volonté de développer les services et le support autour de MySQL. Gardez à l'esprit un mot d'ordre important : TEST, TEST, TEST. Rien ne remplacera les tests pour éprouver les bases et choisir celle qui convient le mieux entre les fonctions voulues et la réalité du terrain. Le maquettage d'un petit projet peut fournir des informations primordiales.

■ F.T.

bases traditionnelles. Autre inconvénient, elle consomme énormément de ressources mémoire.

Le côté développeur

SQL demeure le langage de requêtage par excellence. Bien que normalisé, chaque éditeur peut réaliser une implémentation qui lui est propre et chaque SGBD n'aura pas forcément le même niveau SQL. Cela peut être un point important pour les applications, les données et donc le développeur. Par exemple dans MySQL, on sera limité dans la capacité de création de sous-requêtes. Ensuite, des éditeurs proposent leur propre SQL enrichi, comme le fait

Oracle ou Microsoft. Pour certaines données, il faut aussi disposer de capacités spécifiques comme le Blob (Binary Large Object) pour stocker des données non structurées. Pour des données médicales ou géographiques, il faut faire appel à des types de données spécifiques. Ou encore connaître le niveau de support de XML ou de l'objet. Ensuite, il y a bien entendu les procédures stockées qui facilitent bien la vie des développeurs et DBA. Celles-ci doivent être écrites dans le langage procédural du SGBD (Transac-SQL pour SQL Server ou Adaptive Server Enterprise, PL-SQL pour Oracle...). Ces langages permettent de combi-

ner des requêtes SQL et des instructions procédurales (boucles, conditions...), dans le but de créer des traitements complexes destinés à être stockés sur le serveur de base de données (objets serveur), c'est le cas par exemple des procédures stockées. Ces langages doivent bien évidemment être maîtrisés par le développeur. En cas d'acquisition d'une nouvelle base, ces derniers devront se familiariser avec le nouveau langage. Et il risque d'y avoir des changements lorsque l'on met à jour sa base. Là encore, il faudra bien vérifier cela.

■ **Pejman Tabassomi & Vincent Malbete,**
Consultants Quest Software.

Oracle 11g : les nouveautés SQL et PL/SQL

Si la nouvelle mouture du serveur de données d'Oracle intéressera sans nul doute les administrateurs qui optimiseront plus facilement la gestion de la mémoire et feront remonter des alertes de dysfonctionnement (auto diagnostic, auto administration et auto configuration), certaines nouvelles fonctionnalités SQL et PL/SQL intéresseront les programmeurs.

Colonnes virtuelles

Une colonne virtuelle n'est pas stockée sur le disque mais évaluée automatiquement à la demande (requête ou mise à jour).

Création d'une colonne

Au niveau de la création d'une table, la syntaxe est la suivante :

```
colonne [typeSQL] [GENERATED ALWAYS] AS (expression)
[VIRTUAL] [ contrainteLigne [contrainteLigne2]...]
```

- Le type de la colonne (si vous ne voulez pas qu'il soit automatiquement déduit de l'expression) suit éventuellement son nom.
- GENERATED ALWAYS et VIRTUAL sont fournies pour rendre le code plus clair.
- L'expression qui suit AS détermine la valeur de la colonne (valeur scalaire).

Une colonne virtuelle peut être indexée mais pas directement modifiable. Seule une modification des valeurs qui interviennent dans l'expression fera évoluer une colonne virtuelle. L'exemple suivant déclare une colonne virtuelle (calcul du nombre d'heures de vol par mois).

```
CREATE TABLE Avion(immat CHAR(6), typeAvion CHAR(15),
nbhVol NUMBER(10,2), age NUMBER(4,1),
freqVolMois GENERATED ALWAYS AS (nbhVol/age/12) VIRTUAL,
nbPax NUMBER(3), CONSTRAINT pk_Avion PRIMARY KEY(immat));
INSERT INTO Avion (immat,typeAvion,nbhVol,age,nbPax)
VALUES ('F-WTSS', 'Concorde', 20000, 18, 90);
INSERT INTO Avion (immat,typeAvion,nbhVol,age,nbPax)
VALUES ('F-GHTY', 'A380', 450, 0.5, 460);
```

La description de la table fera apparaître la colonne virtuelle. Pour obtenir des valeurs, il suffit d'évaluer son expression à partir d'une requête.

```
SELECT immat, freqVolMois FROM Avion;
IMMAT  FREQVOLMOIS
-----
F-WTSS 92,59259592...
F-GHTY 75
```

Ajout d'une colonne

```
ALTER TABLE nomTable ADD
colonne [typeSQL] [GENERATED ALWAYS] AS (expression)
[VIRTUAL] [ contrainteLigne [contrainteLigne2]...];
```

Ajoutons à la table la colonne virtuelle qui détermine le ratio du nombre d'heures de vol par passager avec une contrainte en ligne.

```
ALTER TABLE Avion
ADD heurePax NUMBER(10,2) AS (nbhVol/age)
CHECK (heurePax BETWEEN 0 AND 2000) NOT NULL;
```

La figure suivante illustre la table lorsqu'elle est sollicitée en INSERT, UPDATE ou DELETE.

Avion

immat	typeAvion	nbhVol	age	nbPax	freqVolMois	HeurePax
F-WTSS	Concorde	20 000	18	90	92,5925...	1111,11
F-GHTY	A380	450	0,5	460	75	900

Restrictions

- Seules les tables de type heap (par défaut) peuvent héberger des colonnes virtuelles.
- L'expression de définition d'une colonne virtuelle ne peut pas faire référence à une autre colonne virtuelle et ne peut être construite qu'avec des colonnes d'une même table.
- Le type d'une colonne virtuelle ne peut être XMLType, any, spatial, media, personnalisé (user-defined), LOB ou LONG RAW.

Table en lecture seule

Une table peut être déclarée en lecture seule.

```
ALTER TABLE nomTable { READ ONLY | READ WRITE } ;
```

Déclarons la table Avion en lecture seule, puis tentons d'insérer un enregistrement. Après avoir annulé cette restriction, les insertions ou mises à jour sont permises.

```
ALTER TABLE Avion READ ONLY;
INSERT INTO Avion (immat,typeAvion,nbhVol,age,nbPax)
VALUES ('F-NEW', 'A318', 90, 1, 140);
ERREUR à la ligne 1 : ORA-12081: opération de mise à jour interdite sur la table "SOUTOU"."AVION"
```

```
ALTER TABLE Avion READ WRITE;
INSERT INTO Avion (immat,typeAvion,nbhVol,age,nbPax)
VALUES ('F-NEW', 'A318', 90, 1, 140);
1 ligne créée.
```

Expressions régulières

Les fonctions de recherche de chaîne de caractères et d'extraction de sous-chaîne à partir d'une expression régulière sont enrichies d'une option supplémentaire qui permet de cibler une sous-expression particulière de l'expression régulière à évaluer.

Recherche et extraction

```
REGEXP_INSTR (source, modèle [, position [, occurrence [, optionRetour
[, paramètre ] [, sousexpr ] ] ] ] )
```

- source indique la chaîne à examiner (une colonne de type CHAR, VARCHAR2, NCHAR, NVARCHAR2, CLOB, ou NCLOB).
- modèle désigne l'expression régulière (jusqu'à 512 octets).
- position est un entier indiquant la position de début de recherche (par défaut 1).

- *sousexpr* est un entier (de 0 à 9) qui permet de rechercher une position d'une sous-expression régulière (fragment d'expression entre parenthèses). Une sous-expression peut être imbriquée et est numérotée dans l'ordre d'apparition en fonction des parenthèses. Si l'option *sousexpr* est différente de zéro, alors la position de la sous-chaine (fragment) qui correspond à l'ordre de la sous-expression est retournée. Si aucune position n'est trouvée, la fonction retourne zéro.

Considérons "(IUT)(R(ei)(ms))", il vient quatre fragments qui sont respectivement "IUT", "Reims", "ei" et "ms". La requête suivante détermine la position de la troisième sous-expression ("ei") au sein de la chaîne de caractères source (IUTReims).

```
SELECT REGEXP_INSTR('IUTReims', '(IUT)(R(ei)(ms))', 1, 1, 0, 'i', 3)
      "Exemple REGEXP_INSTR" FROM DUAL;
Exemple REGEXP_INSTR
```

5

Concernant l'extraction, on retrouve la même option.

```
REGEXP_SUBSTR (source, modèle [, position [, occurrence [, paramètre ]
[, sousexpr ] ] ] )
```

Si aucune sous-expression n'est trouvée, la fonction retourne NULL. Extrayons la troisième sous-expression.

```
SELECT REGEXP_SUBSTR('IUTReims', '(IUT)(R(ei)(ms))', 1, 1, 'i', 3)
      "Exemple REGEXP_SUBSTR" FROM DUAL;
```

Exemple REGEXP_SUBSTR

ei

Comptage

La fonction `REGEXP_COUNT` compte le nombre d'occurrences d'une expression régulière dans une chaîne de caractères. Si aucune occurrence n'est trouvée, la fonction retourne zéro.

```
REGEXP_COUNT (source, modèle [, position [, paramètre ] ] )
```

- *source position* et *paramètre* ont la même signification que pour `REGEXP_INSTR`.
- *modèle* désigne l'expression régulière, si des sous-expressions sont présentes (fragments), elles seront ignorées et considérées comme un tout.

L'exemple retourne le nombre de fois que l'expression IUT est présente dans la chaîne source.

```
SELECT REGEXP_COUNT('IUTBlagnacIUT', '(IU)T', 1, 'i')
      REGEXP_COUNT FROM DUAL;
REGEXP_COUNT
```

2

Déclencheurs

Activation et désactivation

Avant la version 11g, tout déclencheur créé était de fait actif (enable). Il est désormais possible à l'aide de la directive `DISABLE` de déclarer un déclencheur en le désactivant dès sa création.

```
CREATE TRIGGER [schéma.] nomTrigger
... [ ENABLE | DISABLE ]
```



```
BEGIN
```

```
...
```

```
END;
```

Ordre d'exécution

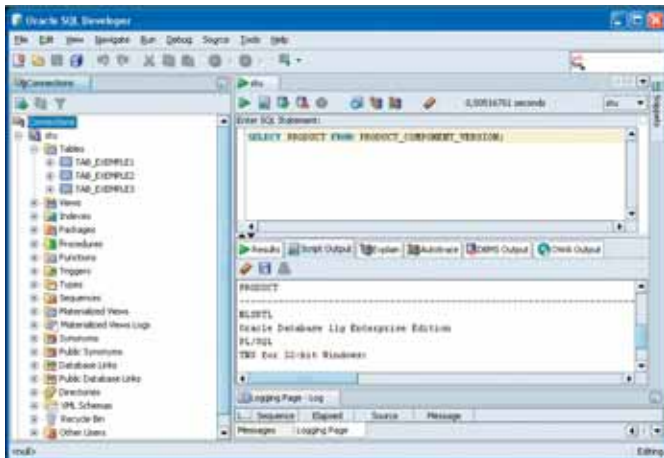
Avant la 11g, il était possible de programmer plusieurs déclencheurs pour le même événement sans maîtriser l'ordre dans lequel les déclencheurs s'exécutaient. La directive `FOLLOWS` précise cet ordre.

```
CREATE TRIGGER [schéma.] nomTrigger
... FOLLOWS [schéma.] nomTriggerQuiS'exécuteAvant ...
BEGIN
...
END;
```

Déclencheur composé

Un déclencheur composé (compound trigger) permet de programmer plusieurs blocs pour différents événements. Le corps d'un déclencheur composé est constitué d'éventuelles variables globales et de blocs PL/SQL (chronologie des événements au niveau de la ligne ou de l'état). La syntaxe simplifiée est la suivante.

```
CREATE [ OR REPLACE ] TRIGGER [schéma.] nomTrigger
FOR { DELETE | INSERT | UPDATE [ OF col1 [, col2 ]... ] }
ON { [schéma.] nomTable | [schéma.] nomVue }
COMPOUND TRIGGER
-- Variables globales
...
BEFORE STATEMENT IS
BEGIN
...
END BEFORE STATEMENT;
AFTER STATEMENT IS
BEGIN
...
END AFTER STATEMENT;
BEFORE EACH ROW IS
BEGIN
...
END BEFORE EACH ROW;
AFTER EACH ROW IS
BEGIN
...
END;
```



```
END AFTER EACH ROW;
END nomTrigger;
```

Dans le déclencheur suivant, un tableau fait office de variable globale et permet de tracer le code après une insertion multiple et une suppression collective.

```
CREATE TABLE TypeAvion (typav VARCHAR2(5),nomtype VARCHAR2(30));
CREATE TABLE TypeAvionBis (typav VARCHAR2(5),nomtype VARCHAR2(30));

CREATE TRIGGER TrigCompose FOR DELETE OR INSERT ON TypeAvionBis
COMPOUND TRIGGER
TYPE typav_tylab IS TABLE OF VARCHAR2(30) INDEX BY BINARY_INTEGER;
tab typav_tylab;
i NUMBER := 0;
BEFORE STATEMENT IS
BEGIN
i := i + 1;
CASE
WHEN INSERTING THEN tab(i) := 'Avant insertion STATEMENT';
WHEN DELETING THEN tab(i) := 'Avant suppression STATEMENT';
END CASE;
END BEFORE STATEMENT;
AFTER STATEMENT IS
BEGIN
i := i + 1;
tab(i) := 'Après STATEMENT';
FOR i IN 1 .. tab.last LOOP
DBMS_OUTPUT.PUT_LINE(tab(i));
END LOOP;
END AFTER STATEMENT;
BEFORE EACH ROW IS
BEGIN
i := i + 1;
tab(i) := 'Avant événement niveau ligne';
END BEFORE EACH ROW;
AFTER EACH ROW IS
BEGIN
i := i + 1;
CASE
WHEN INSERTING THEN tab(i) := :NEW.typav || ' inséré';
```

```
WHEN DELETING THEN tab(i) := :NEW.typav || ' supprimé';
END CASE;
END AFTER EACH ROW;
END TrigCompose;
/
```

En considérant les données suivantes.

```
INSERT INTO TypeAvion VALUES ('A320','Biréacteur Airbus 320');
INSERT INTO TypeAvion VALUES ('A340','Quadriréacteur Airbus 340');
```

La trace de l'insertion multiple dans la table concernée par le déclencheur décrit la chronologie des actions.

```
INSERT INTO TypeAvionBis SELECT * FROM TypeAvion;
Avant insertion STATEMENT
Avant événement niveau ligne
A320 inséré
Avant événement niveau ligne
A340 inséré
Après STATEMENT
2 ligne(s) créée(s).
```

Les principales restrictions de ce type de déclencheurs sont :

- Seuls les déclencheurs LMD (INSERT, UPDATE et DELETE) peuvent être composés.
- Il n'est pas possible de déclarer un bloc d'exceptions.
- Seule la section BEFORE EACH ROW peut modifier une valeur de type NEW.

Tables mutantes

Avant la 11g, il était interdit de manipuler la table sur laquelle se porte le déclencheur dans le corps du déclencheur lui-même (mutating tables, erreur "ORA-04091"). L'exemple suivant décrit la programmation d'un déclencheur qui compte les lignes d'une table après chaque insertion. L'erreur n'est pas soulignée à la compilation mais levée dès la première insertion.

```
CREATE TABLE Trace (evenement VARCHAR2(100));

CREATE TRIGGER TrigMutant1
AFTER INSERT ON Trace FOR EACH ROW
DECLARE
v_nombre NUMBER;
BEGIN
SELECT COUNT(*) INTO v_nombre FROM Trace;
DBMS_OUTPUT.PUT_LINE('Nombre de traces : ' || v_nombre);
END;
/

INSERT INTO Trace VALUES ('Lisez Programmez!');
ERREUR à la ligne 1 : ORA-06512: à "SOUTOU.TRIGMUTANT1", ligne 4
ORA-04088: erreur lors d'exécution du déclencheur 'SOUTOU.TRIGMUTANT1'
```

Le déclencheur composé résoud ce problème.

```
CREATE OR REPLACE TRIGGER TrigMutant1
FOR INSERT ON Trace COMPOUND TRIGGER
v_nombre NUMBER;
BEFORE STATEMENT IS
BEGIN
SELECT COUNT(*) INTO v_nombre FROM Trace;
```



```
END BEFORE STATEMENT;
AFTER EACH ROW IS
BEGIN
    DBMS_OUTPUT.PUT_LINE ('Nombre de traces : ' || v_nombre);
END AFTER EACH ROW;
END;
```

L'événement initial était AFTER INSERT se traduit par un déclencheur composé FOR INSERT contenant les sections BEFORE STATEMENT (pour interroger la table) et AFTER EACH ROW (pour définir l'action).

```
INSERT INTO Trace VALUES ('Lisez Programmez!');
Nombre de traces : 0
1 ligne créée.
INSERT INTO Trace VALUES ('C'est tellement bien.');
```

Types de données

Sous-type SIMPLE_INTEGER

Le sous-type SIMPLE_INTEGER dérive du type PLS_INTEGER. Bien que son domaine de valeurs soit identique à celui de PLS_INTEGER (-2147483648 à 2147483647), il est affecté d'une contrainte NOT NULL et diffère de son prédécesseur du fait de sa robustesse de capacité de dépassement (overflow). L'erreur "ORA-01426 : numeric overflow" n'est plus levée en cas de dépassement en positif ou en négatif d'une variable de type SIMPLE_INTEGER.

Sous-types flottants

Les sous-types SIMPLE_FLOAT et SIMPLE_DOUBLE dérivent respectivement des types BINARY_FLOAT et BINARY_DOUBLE (mêmes domaines de valeurs). Chacun diffère de son prédécesseur du fait de l'existence d'une contrainte NOT NULL. Sans utiliser de ressources gérant la nullité, ces nouveaux sous-types sont plus performants, lors d'opérations, que leurs prédécesseurs, dans un mode opératoire par défaut (PLSQL_CODE_TYPE='NATIVE').

Directive CONTINUE

La directive CONTINUE fonctionne comme celle du langage Java : elle interrompt l'itération en cours et revient au début de la structure (à la condition pour un WHILE, à l'itération suivante pour un FOR ou à l'instruction qui suit le LOOP) pour éventuellement refaire une nouvelle itération.

```
CONTINUE [ étiquette ] [ WHEN condition ];
```

Code PL/SQL	Résultat
DECLARE x NUMBER := 0; BEGIN LOOP -- On arrive ici après CONTINUE DBMS_OUTPUT.PUT_LINE ('Dans la boucle, x = ' TO_CHAR(x)); x := x + 1; IF (x < 3) THEN CONTINUE; END IF; DBMS_OUTPUT.PUT_LINE ('Après CONTINUE, x = ' TO_CHAR(x)); EXIT WHEN (x = 5); END LOOP; DBMS_OUTPUT.PUT_LINE ('Après la structure, x = ' TO_CHAR(x)); END; /	Dans la boucle, x = 0 Dans la boucle, x = 1 Dans la boucle, x = 2 Après CONTINUE, x = 3 Dans la boucle, x = 3 Après CONTINUE, x = 4 Dans la boucle, x = 4 Après CONTINUE, x = 5 Après la structure, x = 5 Procédure PL/SQL terminée avec succès.

En remplaçant le " IF... " dans le bloc par l'instruction " CONTINUE WHEN x<3; ", on obtient le même résultat. Attention, si vous utilisez CONTINUE dans une boucle FOR manipulant un curseur, vous le fermez automatiquement.

Variable de type séquence

Les directives CURRVAL et NEXTVAL ne sont plus limitées aux instructions SELECT, INSERT, et UPDATE. Les expressions séquence.CURRVAL et séquence.NEXTVAL peuvent apparaître à tout endroit du code où une expression de type NUMBER peut apparaître.

```
CREATE SEQUENCE seqAff MAXVALUE 10000 NOMINVALUE;
DECLARE
seq_valeur NUMBER;
BEGIN
seq_valeur := seqAff.CURRVAL;
...
seq_valeur := seqAff.NEXTVAL;
...
END;
```

■ Christian Soutou

Maître de conférences, dirige le département Réseaux et Télécommunications de l'IUT de Blagnac. Il est l'auteur d'ouvrages sur Oracle, MySQL et UML.

e-MAGAZINES

Les anciens numéros, depuis 2003, sont disponibles, en format PDF. vous pouvez également consulter un article seul, ou un dossier.
Tarif unitaire : 3€ pour un numéro, 1€ pour un article ou un dossier (panier minimum : 3 €)



Archives et abonnements sur la Boutique : www.programmez.com

Le renouveau de Ingres !

On entend beaucoup parler de MySQL et de PostgreSQL mais on oublie souvent de citer le SGBDR Open Source le plus mature : Ingres. En effet le développement d'Ingres a été initié en 1974. D'ailleurs, pour la petite histoire, c'est son créateur qui a lancé le projet PostgreSQL, dix ans après avoir créé Ingres...

Depuis 2004 le code source d'Ingres a été publié sous une licence Open Source et il rejoint donc les autres SGBDR libres. A ce titre Ingres est, sinon le SGBDR libre le plus mature, du moins l'un des plus matures ! Pour les développeurs de tous horizons c'est une opportunité de plus d'utiliser un logiciel libre, pour les équipes d'Ingres c'est un renouveau, une nouvelle stratégie, avec beaucoup d'ouverture. Nous allons brosser un rapide portrait des principales fonctionnalités haut niveau d'Ingres avant de présenter comment mettre en place une réplication maître-maître

Les caractéristiques principales

Les principales caractéristiques sont :

- Un SGBDR mature et stable de classe professionnelle
- Une architecture multi-thread autorisant un usage optimal des architectures multi-cœur (SMP Symmetric multiprocessing).
- Support des standards SQL

Des fonctionnalités avancées :

- Réplication avancée
- Clustering / Fail over
- Partitionnement

Pourquoi et quand utiliser Ingres ?

Quel intérêt d'utiliser Ingres plutôt qu'une autre base de données ? Il s'agit d'une base de données robuste qui a fait ses preuves. Ingres est installé chez de nombreux grands comptes et remplit son rôle en silence depuis des années. Pour un public de DBA ou de développeurs habitués aux outils Open Source, Ingres apportera un réel plus en terme de logiciels annexes : les outils visuels d'administration, de manipulation, d'audit et de développement sont nombreux, bien pensés et bien documentés. En terme d'utilisation, on peut remarquer qu'Ingres est une solution particulièrement bien adaptée à un contexte avec beaucoup d'écritures.

Introduction à la réplication de bases

Concepts et définitions

La réplication permet de synchroniser deux ou plusieurs serveurs Ingres tout en conservant l'état des données identique. La réplication de données entre bases repose sur le module 'Replicator'. Ce module fonctionne de manière asynchrone : il écoute les transactions et les réplique lorsqu'elles sont validées.

En pratique dans quel cadre utilise-t-on la réplication ?

- Pour partager la charge des requêtes de lecture
L'idée classique est de centraliser les requêtes d'insertion sur un serveur maître et de diriger toutes les requêtes de lecture sur un ou plusieurs serveurs esclaves. Ainsi on augmente la scalabilité en lecture.
- Redondance :
Permet la redondance de serveur " fail over " en cas de panne d'un des maîtres. Utilisé pour obtenir de la disponibilité 24/7
- Sauvegardes :
Permet de faire des sauvegardes sans affecter les opérations du serveur maître

Autre point intéressant Il est possible de choisir la 'route' que les données de réplication vont emprunter. Cette fonctionnalité prend tout son sens dans le cas de réplication en cascades ou les débits disponibles entre serveurs ne sont pas forcément les mêmes.

Lorsque l'on met en œuvre une réplication avec Ingres, certains termes et fonctionnalités sont à connaître :

- Les **Collisions** : elles apparaissent lorsque deux utilisateurs modifient la même ligne au même moment à deux endroits différents. Ces collisions peuvent être détectées et traitées par Replicator ou manuellement. Typiquement, il est possible de choisir la gestion humaine : on pourra alors choisir quelle est la 'bonne' version de la donnée ou encore automatique : Replicator donnera la priorité à une localisation plutôt qu'à une autre.
- Les **CDDs** (Consistent Distributed Data Set) : un CDDs définit précisément quelles données doivent être répliquées et vers quelle destination.
- Les '**CDDs Target Types**' : il existe 3 types de cibles de réplication.
 1. Le serveur maître en écriture (à utiliser comme type de serveur source ou lors d'une réplication multi-maître) ;
 2. Le serveur esclave intelligent qui reçoit des données mais peut détecter les collisions avec les données locales ;
 3. Le serveur esclave simple qui reçoit des données et ne détecte pas les collisions locales (par exemple un esclave en lecture).
- Les **Data Propagation Path** : les chemins de propagation de données permettent de définir, dans chaque CDDs, d'où viennent, par où passent et où vont les données. On y trouve trois notions :
 1. la base source (celle d'où proviennent vraiment les données),
 2. la base locale (généralement la même que la base source, sauf s'il s'agit de créer un chemin en cascade, auquel cas les serveurs intermédiaires se distinguent)
 3. la base destination.

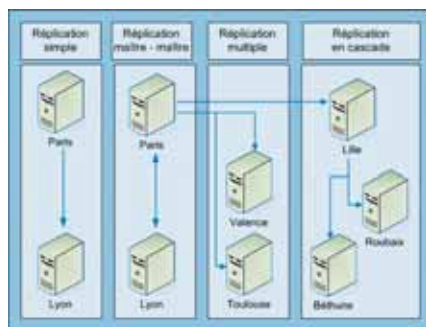


Figure 1 : les différents modes de réplication possibles

Avec la réplication d'Ingres il est possible de choisir les données qui seront répliquées :

- toute une base
- simplement les tables voulues,
- simplement certaines colonnes
- simplement des lignes correspondant à un critère spécifique

Mise en pratique

Nous allons illustrer la réplication en détaillant la mise en place du mode de réplication multi-maître. Pour cela nous utiliserons la base de démonstration installée par défaut avec Ingres 2006 – demodb. Si vous voulez suivre pas à pas notre mise en œuvre, rendez-vous sur le site www.ingres.com, téléchargez et installez Ingres 2006 dans le répertoire " C:/ingres/ " en prenant garde de bien installer le module Replicator (qui n'est pas installé par défaut).



Figure 2 : installation d'Ingres

Test de connexion

Visual DBA est la principale application qui permet d'administrer un serveur Ingres. Une fois lancée, connectons nous au serveur local (Clic droit sur le nœud 'local' puis 'Connect'). La base de données de démonstration doit apparaître sous le dossier 'Databases' du serveur local.

Création d'une seconde base de données

Créons une seconde base – demorep - qui sera en multi-maître avec demodb. Afin de permettre à tous de tester cette réplication, nous allons installer les deux bases sur la même machine (même serveur, même installation). Nous verrons que cela nécessitera tout de même de créer 2 instances Replicator (une pour chaque source de données).

1. Création de demorep

Commençons par créer demorep de façon à ce qu'elle soit identique à demodb. Pour cela, nous utiliserons la commande `relocatedb`. Lancez " Ingres Command Prompt " (Sous Windows, cliquez droit sur l'icône Ingres dans la barre des tâches) et indiquez à Ingres de copier demorep :

```
<code>
relocatedb -new_database=demorep
</code>
```

Pour tester, vous pouvez lancer " Ingres Visual DBA" et vérifier que la base demorep existe bien.

2. Vérification de l'existence du VNode du serveur local

Avant de continuer, nous devons prendre en compte le fait qu'Ingres Replicator utilise les couches réseaux d'Ingres. Or, par défaut, seul le nœud 'local' est déclaré – ce qui n'est pas suffisant. Créons donc un vnode qui pointe vers notre machine comme indiqué sur la figure 3.

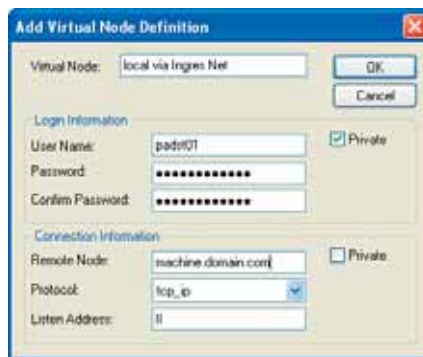


Figure 3 : le VNode

A chaque fois que nous aurons besoin de spécifier une référence à une base 'distante', nous utiliserons ce vnode plutôt que le nœud local.

Activons à présent la réplication sur chacune d'elles. La première chose à faire est d'ajouter les objets au catalogue de chaque base. Il suffit pour cela d'utiliser le menu contextuel 'Install Replication Objects' sur le dossier 'Replication' de chaque base.

A la fin de chaque installation, Visual DBA demande automatiquement de renseigner un certain nombre d'informations. Il s'agit en fait des informations d'identification du serveur et de la base dans le cadre d'une infrastructure de réplication comme le montre la figure 4.

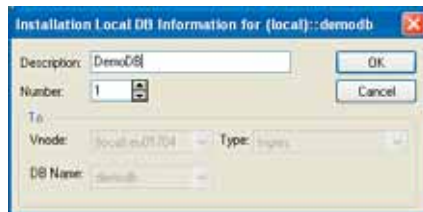


Figure 4 : identification du serveur

Attention, le numéro d'identification '1' est correct pour la base DemoDB, mais il en faut un autre pour la base répliquée comme l'indique la figure 5.



Figure 5 : identification du second serveur

Il nous faut maintenant effectuer l'opération similaire pour déclarer pour chaque base la base distante. Ainsi sur Demo DB, nous déclarons DemoRep dans les 'Databases' disponibles dans le dossier 'Replication' – noter l'utilisation du vnode 'réseau' comme le montre la figure 6.

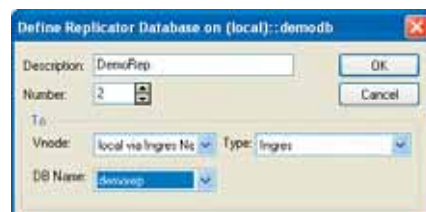


Figure 6 : Déclaration du Vnode réseau

Inversement, sur DemoRep, déclarons DemoDB (cf figure 7):

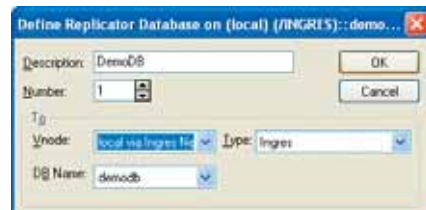


Figure 7 : Déclaration du Vnode réseau

A présent, déclarons le contenu et le sens de réplication au sein d'un CDDS, que nous allons là aussi créer avec les mêmes données d'identification pour chaque base : Toujours dans le dossier " Replication ", créons un nouveau CDDS identique à la figure 8 en suivant les points suivants :

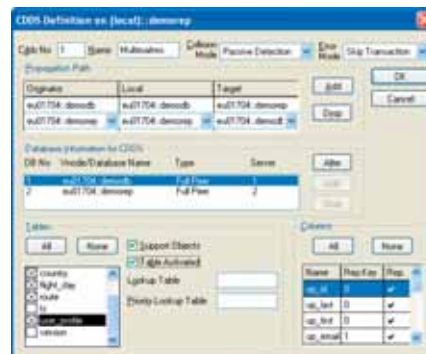


Figure 8 : Définition des CDDS

- donnons lui un nom, par exemple 'Multi-maitre' ;
 - Explicitons les chemins de réplication : nous souhaitons faire une réplication multi-maitres, donc créons les 2 chemins DemoDB ↔ DemoRep et DemoRep ↔ DemoDB. En l'occurrence la colonne 'local' est la même que la colonne 'Originator' car il ne s'agit pas de réplication en cascade.
 - Les Nœuds concernés (zone 'Database Information for CDDS') est remplie automatiquement. Notez les Numéros de serveurs de réplication.
 - Déclarons les tables concernées par la réplication :
 - Il faut activer les tables une à une et, surtout, ne pas oublier de cocher 'Support Objects' et 'Table Activated' table par table, afin d'activer la réplication et la détection de collisions.
 - Les tables 'tz' et 'version' ne pourront être répliquées : en effet, elles ne disposent pas de clé primaire et/ou unique nécessaire à la mise en œuvre de la réplication.
- Ce n'est pas gênant, car ces tables sont statiques. Le fait d'avoir copié la table est donc suffisant

Effectuer l'opération de création du CDDS N°1 pour les 2 bases. A chaque fois, VisualDBA procède à la création des objets et à l'activation de la réplication par table, on passe donc quelques écrans du type :



Figure 9 : Activation de la réplication

Il nous reste à présent à activer 2 services pour notre serveur : il s'agit des services qui vont écouter la base source d'une transaction pour ensuite l'envoyer vers la base cible de la réplication. Même si nos deux bases sont sur la même instance, il nous faut en créer 2. sous Windows, la création et l'installation des services de réplication se fait au moyen de la ligne de commande 'repinst', comme indiqué sur la figure 10.

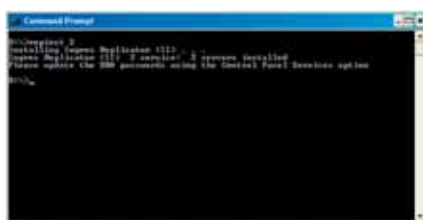


Figure 10 : Création des services de réplication

Le '2' spécifié en argument de la commande nous permet de créer les 2 services. Attention, bien lire ce qui suit l'installation : les modules de réplication sont des outils qui vont écouter les transactions comme s'il s'agissait d'utilisateurs DBAs de nos bases. En l'occurrence, nos bases de données ont été créées par l'utilisateur qui a fait l'installation (rappelez-vous, l'installation a créé DemoDB !). Nous devons donc spécifier le même utilisateur/mot de passe pour les 2 services au sein du gestionnaire de services Windows, comme le montre la figure 11.

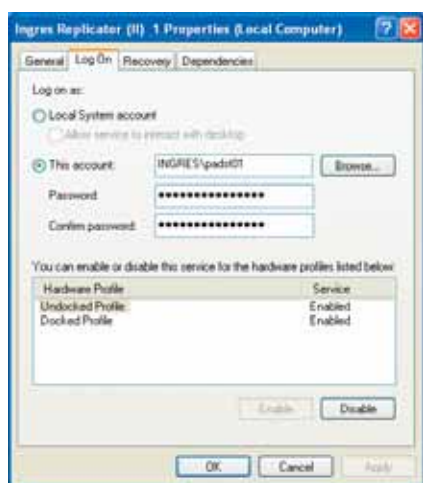


Figure 11: Gestionnaire des services

On pourra en profiter pour démarrer automatiquement les services ... Une fois les services démarrés, nous sommes prêts. Il ne reste plus qu'à tester. Par exemple, reprenons l'application de démo et créons un nouveau profil utilisateur. Insérons les données directement en SQL ou alors on peut utiliser une application de test visuel mis à disposition par Ingres (cf remarque)

Remarque

Une interface graphique utilisant la base de démonstration est disponible avec le plug-in 'Eclipse DTP'. Vous pouvez le télécharger sur

le site d'Ingres. Le plus simple est de télécharger le package 'Eclipse DTP bundle', et de le décompresser sur son disque – il peut être directement lancé.



Figure 11 : Application graphique utilisant la base de démonstration

Par défaut, l'application se connecte sur DemoDB (tant qu'elle est créée sur le serveur local et, bien entendu, sur l'instance 'II' ou encore 'II7' du point de vue JDBC ...) On peut vérifier via Visual DBA que l'utilisateur existe dans DemoDB, comme le montre la figure 12.

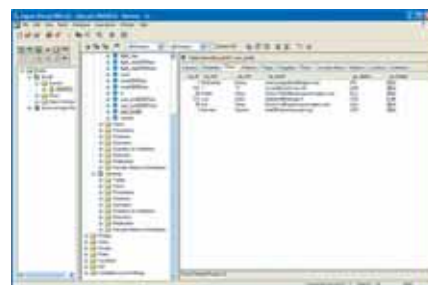


Figure 12 : Visual DBA

Validons également l'aspect bi-directionnel des mises à jour. Pour cela, nous pouvons par exemple, connecter l'application de démo sur la base de démo répliquée, puis modifions l'information de notre profil – vérifions alors que DemoDB est également modifiée.

■ Stéphane PADIQUE

Responsable avant-vente Europe du sud pour Ingres.

■ Cyril PIERRE de GEYER

Responsable du pôle R&D de la société Anaska spécialisée dans les formations relatives aux technologies Open Source. Co-auteur des livres PHP5 avancé et PHP & SQL.

SQL SERVER 2008 : du développeur au DBA

Voici de nombreuses années que je travaille sur SQL Server. La 6.5, la 7 et la version 2000 se succédèrent gentiment. Puis... vint la sortie de SQL 2005 et la je fus... surprise par ses nombreuses évolutions, plus de puissance, plus de robustesse, une offre BI intégrée et j'en passe. Mais j'ai été étonnée par la volonté clairement affichée de Microsoft d'orienter son produit vers le développeur. La découverte du management studio dans les premières bêtas de " Yukon " loin des classiques consoles d'administration MMC avait été un révélateur.

Cet environnement 2005 était taillé sur mesure pour le confort du développeur auparavant condamné à passer de son sacro saint Visual Studio au rudimentaire Query Analyzer. Là, il pouvait même l'espace d'un instant –le temps de réaliser l'incongruité complète de la chose (1) - rêver au Managed Code et imaginer oublier le TSQL... Bref, le tapis rouge était déroulé pour amener, ramener le développement dans le cocon SQL. Katmai- aka SQL SERVER 2008, lui au contraire a directement appuyé sur la corde sensible du DBA. Un sacré changement s'annonçait pour nous.

Nous, c'est à dire l'administrateur de base de données. Vous savez ? Cet homme obscur dans les soutes du serveur. Ce DBA qui s'arrache régulièrement les cheveux : voir son serveur malmené par des requêtes maladroitement formulées (2), des approches inefficaces, des index mal posés, des bases de données tellement mal modelées que l'on peut sans craindre parler d' " abnormal form ".

Il faut admettre que le DBA dans le monde de SQL Server est un élément somme toute peu répandu. Très souvent même simplement inexistant. SQL ,ce système de base de données qu'on utilise, ou plutôt utilisait " à côté " des gros mamouths, ce système qui fonctionne " out of the box ", avec sa mémoire " dynamique ", son locking " dynamique ", son " ça marche tout seul, laissez l'optimiseur faire " (3). Un DBA pour ÇA ? Inutile, le Monsieur/Madame qui gère l'Exchange et le réseau et toutes ces choses saura bien s'occuper de configurer les quelques jobs nécessaires aux backup, il y a un wizard pour les plans de maintenance...et quoi d'autre ?

Mais voilà... voilà qu'insidieusement SQL Server a envahi les parcs. Utilisé en back end aussi bien par les développements maison que par les logiciels. Voilà qu'il tourne en 64 bits (4), voilà qu'il rivalise avec les autres...

L'administrateur SQL jusqu'à présent utilisait les maigres outils de base fournis avec le produit et travaillait artisanalement avec le plus de bon sens possible. Bref, des recettes de sorciers qui tentaient à coups de scripts, de traces, de surveillance étroite de juguler le flot des curseurs voraces et d'updates en masse. Avec SQL 2008 est venu le temps de l'industrialisation et de l'efficacité. Je vais donc m'attacher à explorer dans cet article deux de ces nouveautés phares.

Le Ressource Governor et le Declarative Management Framework (DMF)

Le Resource Governor est un fantôme devenu réalité dans la vie du DBA puisque il va permettre tout simplement d'assigner à chacun les ressources qu'il pourra utiliser et dans quelles proportions sur le serveur.

Petit rappel : quelles sont les ressources que le serveur SQL aime à utiliser ?

C'est la **mémoire** et le **processeur**. Concernant la première c'est très simple, SQL Server tendra à utiliser toute la mémoire qui lui est laissée disponible. Pour la simple et bonne raison qu'il est plus rapide de lire une page en mémoire que de faire un accès disque. Un serveur SQL en bonne santé doit avoir un ratio de pages en cache au dessus des 90%. On ne configure en général pas la mémoire d'un serveur SQL, tout au plus on la bride si le serveur n'est pas dédié (ce qui est rarement une bonne idée) ou si l'on se trouve dans un contexte de serveurs en clusters actif/actif (5). Le processeur, lui, est aussi très fortement sollicité et une requête boiteuse est capable, hélas, de l'emballer bien loin...Finalement, ce Resource Governor aura un usage conceptuellement semblable à celui des applications pools de IIS. Isolation et protection.

Comment cela fonctionne-t-il ?(6)

Le principe est assez simple on va travailler avec 3 éléments :

- Workload Group
- Resource Pool
- Function Classifier.

On pourra, bien sûr, travailler via l'explorateur d'objets du Management Studio ou directement en T-SQL.

Le Resource Governor se trouve dans la partie " Management " de l'explorateur d'objets. (Fig.1) (Fig.2).

Workload Group - Un Workload Group est un contenant utilisé pour limiter certaines ressources spécifiques du Serveur SQL telles que le Degré de parallélisme ou les requêtes utilisateurs.

Resource Pool - Un Resource Pool est une collection de ressources systèmes comme la mémoire ou le CPU.

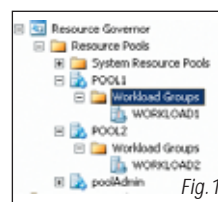


Fig.1

(1) Il est évident que le TSQL reste le langage de manipulation de données le plus efficace, le Managed code apportant simplement la possibilité d'apporter un business logique complexe au cœur même du process SQL et de bénéficier de sa stabilité. (2) Sans vouloir d'emblée vexer mon honorable lectorat, force m'est de constater après 12 années de pratique que le result set semble le seul but, peu importe le chemin et l'IO pour l'atteindre. Hélas ! (3) Au point que dans les formations officielles de Microsoft que j'ai données tant de fois, il est clairement recommandé d'éviter tout " hint " et autre forçage de main dans les requêtes entre autres. (4) Pour être tout à fait exhaustive, depuis la version 2000 ! (5) Attention s'entend dans le sens de Microsoft, à savoir un serveur de fault tolerance et non d'équilibrage de charge. Le but étant d'assurer la possibilité aux deux instances virtuelles de travailler sur un seul nœud en cas de panne. (6) Les exemples et codes donnés ici sont repris et inspirés de l'excellente e-démo de Bertrand Audrass voir <http://www.microsoft.com/france/vision/WebcastTechnet.aspx?EID=17431ccf-4f01-44d3-bda2-e228491884aa>

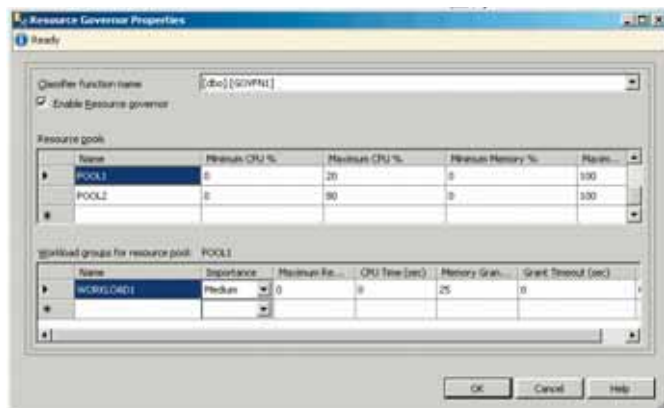


Fig.2

Un Workload Pool est assigné à un Workload Group, qui lui est assigné au Resource Governor via la fonction de classification.

Configuration des Resource Pools

Créer des Resource Pools consiste à configurer les ressources minimum et maximum d'utilisation du CPU et de la mémoire. En gros, nous pouvons trancher le "gâteau" du serveur en portions plus ou moins congrues. Ici je crée deux pools aussi gourmands l'un que l'autre. On peut reconfigurer ces ressources au vol, ce qui permet de dynamiquement favoriser Pierre ou Paul sans devoir redémarrer le service et ça... C'est beau !

```
CREATE RESOURCE POOL pool1
WITH
(
    MAX_CPU_PERCENT=100,);
Go
CREATE RESOURCE POOL pool2
WITH
(
    MAX_CPU_PERCENT=100,);
```

La configuration des Workload Groups

Le Workload group lui va vous permettre de définir un ensemble de critères et d'y associer un Resource pool. Le code ci-dessous crée deux Workload group "workload1" et "workload2" :

```
CREATE WORKLOAD GROUP [WORKLOAD1]
WITH
( group_max_requests=0,
  importance=Medium,
  request_max_cpu_time_sec=0,
  request_max_memory_grant_percent=25,
  request_memory_grant_timeout_sec=0,
  max_dop=0
)
USING [POOL1]
GO
CREATE WORKLOAD GROUP [WORKLOAD2]
WITH
( group_max_requests=0,
  importance=Medium,
```

```
request_max_cpu_time_sec=0,
request_max_memory_grant_percent=25,
request_memory_grant_timeout_sec=0,
max_dop=0
)
USING [POOL2]
GO
```

Dans ces lignes de code, nous précisons le degré de "criticité" du groupe comme étant "medium", ce qui est la valeur par défaut. Nous pouvons comparer cela au paramètre "priorité" du système d'exploitation pour un execution thread.

Nous limitons également chaque requête à 25 % de la taille mémoire du Pool, cela signifie qu'aucune requête de ce groupe ne peut consommer à elle seule plus de 25% des ressources mémoire allouées à ce groupe. Nous paramétrons ensuite le temps d'exécution et d'attente à 0 (infini) et le degré de parallélisme pourrait également être configuré. Tout ceci, en affectant d'emblée le Workload group à son pool de ressources.

Créer une fonction de classification

Pour permettre à SQL Server d'utiliser les Workload Groups que vous venez de créer, il va falloir classer chaque connexion au serveur pour déterminer à quel groupe appartient chacune. Cela se fait en utilisant une "fonction de Classification". Ces fonctions sont neuves dans SQL server 2008 et s'exécutent à chaque nouvelle connexion au serveur. Ce sont des fonctions scalaires user-defined et elles sont basiquement utilisées pour retourner le nom du Workload Group, cible de chaque connexion utilisateur. Le code ci-dessous crée une fonction assez simple pour simplement se baser sur le nom de l'utilisateur et le renvoyer dans la "bonne équipe". Evidemment, hors d'un cadre de démonstration il sera beaucoup plus pertinent de s'appuyer sur le nom de l'application utilisée, ou pourquoi pas de l'host etc. Le savoir-faire de l'administrateur ainsi que sa connaissance fine de l'utilisation du serveur l'aidera à séparer le bon grain de l'ivraie.

```
CREATE FUNCTION GOVFN1()
returns sysname with schemabinding
begin
declare @val sysname
if 'user1'=user_name() set @val='workload1'
if 'user2'=user_name() set @val='workload2'
return @val
end
```

N'oublions pas de donner accès à cette fonction :

```
Grant execute on dbo.GOVFN1 to public
```

Assigner la fonction au Resource Governor

L'étape finale de cette configuration sera d'assigner et d'activer la fonction de classification. Le grand "départageur" sera prêt à veiller sur les ressources de votre SQL comme on pourra l'observer.

```
ALTER RESOURCE GOVERNOR WITH (CLASSIFIER_FUNCTION=dbo.
GOVFN1());
Go
ALTER RESOURCE GOVERNOR RECONFIGURE;
```


A présent, nos utilisateurs user1 et user2 seront impitoyablement "cadrés"

Vérifier le fonctionnement

Une petite requête sur une des fameuses vues dynamiques dont SQL2005 nous a donné le goût, et qui bien sûr nous sont proposées dans un éventail encore plus large en 2008, nous permettra de visualiser les connexions en cours et leur classification

```
Select ses.session_id,ses.group_id,gp.name
from sys.dm_exec_sessions as ses
join sys.dm_resource_governor_workload_groups as gp
on ses.group_id=gp.group_id where ses.session_id >50
```



Fig.3

Nous pourrions ensuite aller plus loin en vérifiant l'usage des ressources via le moniteur système des ressources ! On lance "perfmom" et on va rechercher les compteurs ad-hoc (Fig.3).

Il nous reste à présent à simuler l'activité de Mr User1 et Mr user2 sur notre serveur. Quelques lignes fatigantes en se connectant sous l'identité du premier :

```
set nocount on
declare @i int =100000000
declare @s varchar(100)
while @i>0
begin select @s=@@VERSION
set @i=@i+1
end
```

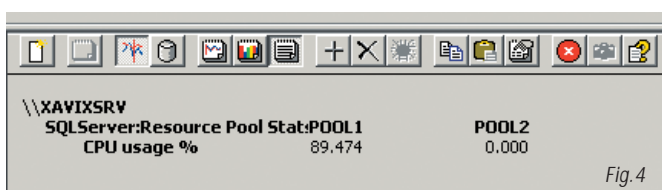


Fig.4

Et on réitère ensuite sous l'identité du 2e :

```
set nocount on
declare @i int =100000000
declare @s varchar(100)
while @i>0
begin select @s=@@VERSION
set @i=@i+1
end
```

Et il reste à admirer de belles courbes, anguleuses certes, mais sous contrôle ! Et ça on adore ! (Fig.5).

Pour conclure cette partie j'ajouterai que les BOL offrent une bonne approche et méthode pour cerner la " fréquentation " de votre serveur et commencer en traçant cette activité avant ensuite de placer des pools et des groupes selon affinités et priorités. Avec l'ingénieuse et

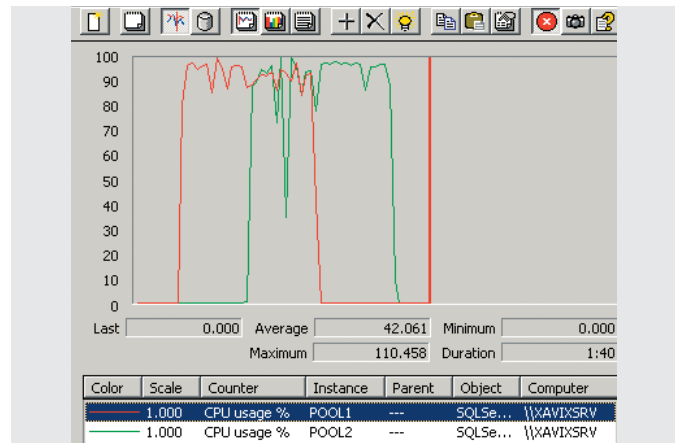


Fig.5

" must to have " idée de bien entendu réserver aux sysadmin suffisamment de ressources en permanence pour veiller au bien-être de tout ce petit monde.

L'administration par Règles

Le deuxième élément abordé ici est le fameux " déclarative management Framework ". Enfin je puis faire obéir mon serveur à un ensemble de règles que je définis. A nouveau, un parallèle simple pour illustrer le concept. On se retrouve ici face à l'équivalent des Politiques de l'Active Directory. On va pouvoir créer un ensemble de règles à observer et de conditions selon lesquelles ces règles se doivent d'être appliquées sur notre serveur SQL. Jusqu'à présent l'approche que nous avions était plutôt " réactive " Nous passons le BPA (Best Practice Analyzer Tool) sur nos serveurs pour détecter et corriger tout manquement aux règles de bonne conduite. Des procédures mal nommées, des bases de données en mode full sans sauvegarde du journal de transaction, des trous de sécurité etc. Peu d'outils nous permettaient de contrôler a priori la façon dont DEVAIENT se dérouler les choses, configuration, création d'objets, tout était soumis à l'aléatoire et à la faillible interface chaise clavier. Seuls les DDL triggers (lourds) pouvaient amener un petit filet de sécurité sur certaines choses. Maintenant, nous pouvons de façon très confortable et efficace établir notre charte de travail. Le mode d'emploi est intuitivement très simple. Trois termes apparaissent :

Politiques/Conditions /Facets

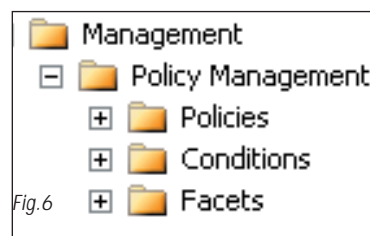


Fig.6

C'est via la partie management du Studio que l'on pourra travailler (Fig.6).

Microsoft fournit déjà une *foultitude* de politiques à activer, issues en droite ligne notamment de son Best Practice Analyzer en voici un

aperçu non exhaustif (Fig.7). Si on regarde d'un peu plus près, on verra qu'une Policy est tout simplement la vérification d'une condition sur un ensemble de cibles (Fig.8). Par exemple, on va vérifier la séparation physique des fichiers de sauvegardes des fichiers de la base de données. On peut choisir quelles seront les bases de données cibles de cette Policy. On définira ensuite le mode d'exécution : à la demande ou planifié (Fig.9). Enfin, on pourra y ajouter un " server restriction " pour tra-



Fig.7

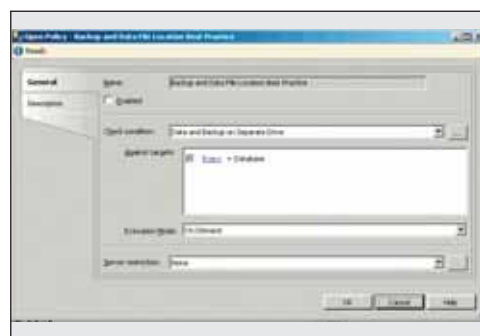


Fig.8

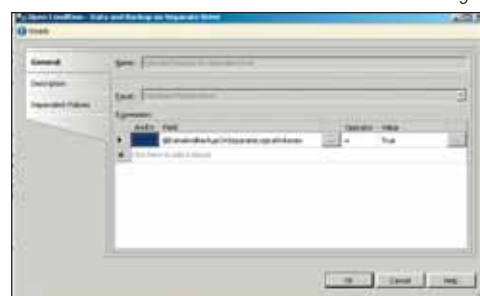


Fig.11

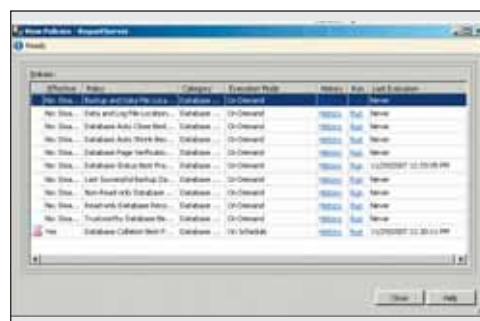


Fig.14



Fig.9



Fig.10

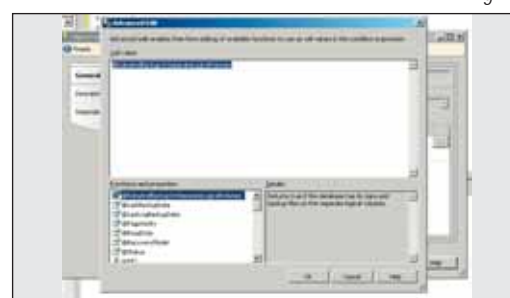


Fig.12

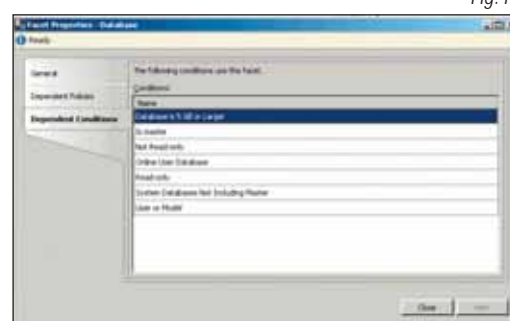


Fig.15

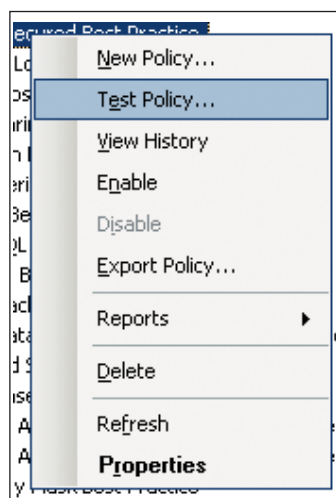


Fig.13

vailler plus finement si nécessaire (Fig.10). Les conditions, elles, s'appuient sur une série de fonctions mises à true ou false (Fig.11). Il est parfaitement loisible d'y ajouter ses propres souhaits (Fig.12). Pour autant que l'offre de départ ne suffise pas !!! Puisqu'il y a quelques dizaines de possibilités déjà présentes et qui couvrent largement les besoins standards d'administration.

Comment cela fonctionnera-t-il ?

Selon le mode d'exécution choisi, les politiques donneront naissance à des triggers DDL, des event notifications ou des jobs. On peut se placer dans une situation où l'on choisit de vérifier à la demande l'adéquation à une règle (Fig.13). Il est possible également pour un objet donné (une database par exemple) de voir toutes les règles qui y sont potentiellement applicables et si ces règles sont activées ou ont été vérifiées un jour (Fig.14). Ceci me mène tout droit à la dernière notion qui est celle des "Facets". Le "facet" est un set de propriétés logiques qui sont liées à une zone de management. Par exemple il y a un facet "database security" qui regroupe les règles et conditions à observer pour assurer au mieux cette "zone" de gestion (Fig.15). Son intérêt est d'être exportable et réalisable à l'envi sous forme de fichier XML.

Petite note : il est évident que toutes ces configurations seront sauveées dans MSDB, la base de données qu'il convient de ne pas négliger dans

nos backup sous peine de perdre nos efforts. Pour conclure sur les Policy, nous voici donc dotés d'un système extrêmement souple et complexe. Un outil autant d'audit que de prévention. A mes yeux, un des mérites dérivés du DMF est tout simplement la sensibilisation à des points qui jusqu'à présent restaient peu connus, peu maîtrisés face à l'apparente simplicité du serveur SQL.

Quelques infos utiles

- <http://www.microsoft.com/sql>
- <http://www.codeplex.com/>
- Technical Communities, Webcasts, Blogs, Chats & User Groups
<http://www.microsoft.com/communities/default.msp>
- Microsoft Learning and Certification
<http://www.microsoft.com/learning/default.msp>
- Microsoft Developer Network (MSDN) & TechNet
<http://microsoft.com/msdn>
<http://microsoft.com/technet>
- Trial Software and Virtual Labs
<http://www.microsoft.com/technet/downloads/trials/default.msp>

■ Isabelle Van Campenhoudt

Data & BI Branch Manager

email: ivancampenhoudt@ict7.com

Abonnez-vous



1 ÉCO

Recevez
le magazine
chaque mois

économisez **20 €**

11 Numéros

45 €

Au lieu de 65,45 €
(Prix au numéro)

(Prix France
métropolitaine)

-40%

2 Étudiant

Vous devez
justifier de votre
statut d'étudiant

Economisez **26 €**

11 Numéros

39 €

Au lieu de 65,45 €

(Prix au numéro)

(offre réservée
France
métropolitaine)

-30%

3 Numérique

Lisez
chaque mois
le magazine

Format PDF
(téléchargement)



11 Numéros

30 €

Tarif Monde entier

Inscription
en ligne
uniquement

2,13 €
le numéro

www.programmez.com

+ Abonnement INTÉGRAL

NOUVEAU

ACCÈS ILLIMITÉ aux ARCHIVES du MAGAZINE pour **1 €** par mois !

Prix de lancement

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

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

OUI, je m'abonne ! ou abonnement en ligne : www.programmez.com

- ☐ **ABONNEMENT 1 an ECO** au prix de 45 € TTC. Tarif France métropolitaine.
Tarifs hors France métropolitaine : CEE et Suisse : 51,83 € - Algérie, Maroc, Tunisie : 55,95 € - Canada : 64,33 € - Tom : 79,61 € - Dom : 62,84 € - Autres : nous consulter
- ☐ **ABONNEMENT 1 an ETUDIANT (11 n°)** : 39 € TTC. Offre limitée à la France métropolitaine. Photocopie de la carte d'étudiant obligatoire
- ☐ **+ SUPPLEMENT ABONNEMENT 1 an INTEGRAL** au prix de lancement de 12 € TTC. (s'ajoute à une des formules d'abonnement)
- MONTANT TOTAL DE L'ABONNEMENT : €

☐ 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 - 75010 Paris.
abonnements.programmez@groupe-gli.com

Programmez!
LE MAGAZINE DU DÉVELOPPEMENT

Offre limitée,
valable jusqu'au
29 février 2008

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

Conformément à la loi Informatique
et Libertés du 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.

Une galerie de Photos sur le Web avec HOP

HOP est une nouvelle plate-forme de développement conçue pour faciliter le développement d'applications interactives sur le Web. Dans le précédent article consacré à HOP (Programmez! N° 104) plusieurs courts exemples ont été présentés qui illustraient la simplicité de développement et de mise en œuvre.

Ce second article montre que cette simplicité est préservée même lorsque les applications croissent en taille. Nous allons présenter une application multimédia, simple mais réaliste, pour afficher des photographies. Son implantation complète (une cinquantaine de lignes de code) est si compacte qu'elle tient intégralement dans cet article.

Une application multimédia sur le Web avec HOP

HOP dispose de nombreuses bibliothèques d'exécution qui facilitent des opérations aussi variées que l'extraction d'informations contenues dans un fichier MP3, la connexion à un serveur de mails, l'ajout de nouveaux événements dans un calendrier ICAL ou l'analyse du trafic DBUS d'une machine Linux. Dans cet article, nous allons utiliser deux de ces bibliothèques pour réaliser une galerie de photos sur le Web. Notre application affichera, pour chaque photo, une vignette accompagnée de la date à laquelle la photo a été prise. Lorsque l'utilisateur cliquera sur une de ces vignettes, une animation graphique fera apparaître la photo correspondante avec un effet de zoom et de rotation.

Rappelons qu'un programme HOP s'exécute toujours, par coopération entre deux processus pouvant s'exécuter sur deux ordinateurs distincts ou sur un même ordinateur. Le premier de ces processus est le courtier, c'est-à-dire un serveur Web léger qui embarque les compilateurs et interprètes HOP et qui permet l'exécution des programmes (ou weblets) HOP. Le second processus, le client, est un simple navigateur.

Dans l'application présentée dans cet article, le courtier lira les fichiers de photos et extraira les informations contenues, le client les affichera au moyen d'un élément canvas.

L'élément CANVAS

Les canevas de HOP reposent sur une extension d'HTML 4.01 apportée par la spécification WhatWG Web applications 1.0 qui a été reprise par le groupe de travail en charge d'HTML 5 (<http://www.w3.org/html/wg/html5>). Les canevas constituent des zones de dessin affichées dans le navigateur Web qui permettent de réaliser des effets visuels et des animations. Avec HOP, les canevas peuvent être indifféremment utilisés avec les quatre principaux navigateurs Web : Firefox, Safari, IE 7 et Opera. Dans notre application, ils seront utilisés pour afficher et animer les photographies. Comme en HTML, un canevas se crée au moyen de la balise <CANVAS> :

```
1: (define canvas
2: (<CANVAS>
3:   :width ~(- (current-window-width) 200)
4:   :height ~(- (current-window-height) 20)))
```

Ces quatre lignes de programme définissent un canevas qui est affecté à une variable globale du courtier nommée canvas. Le canevas ne sera

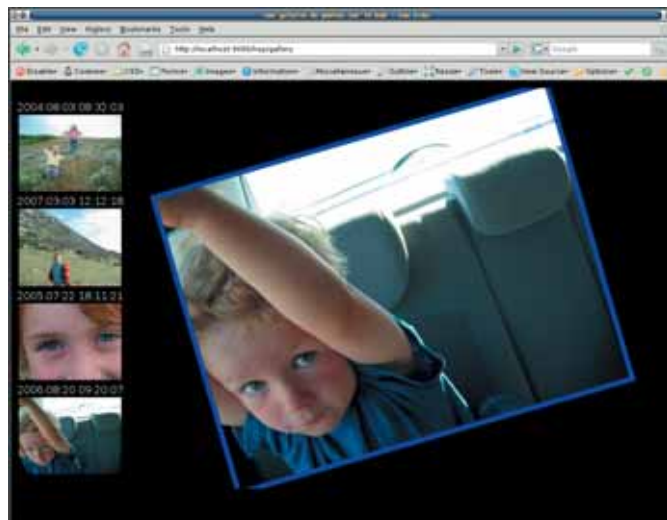


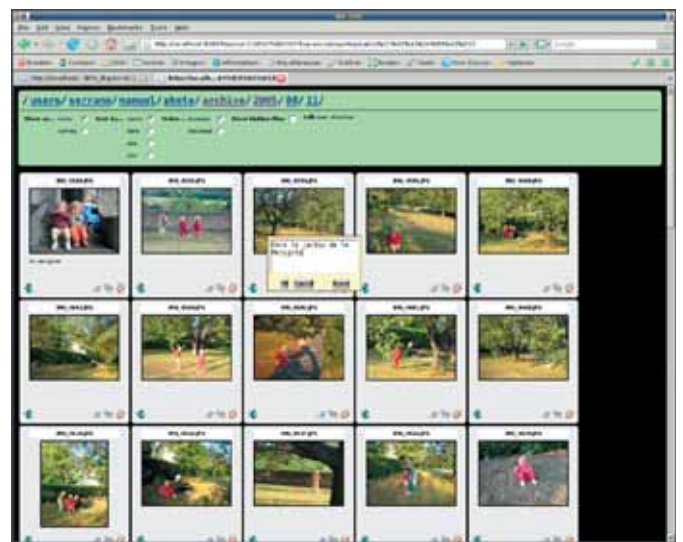
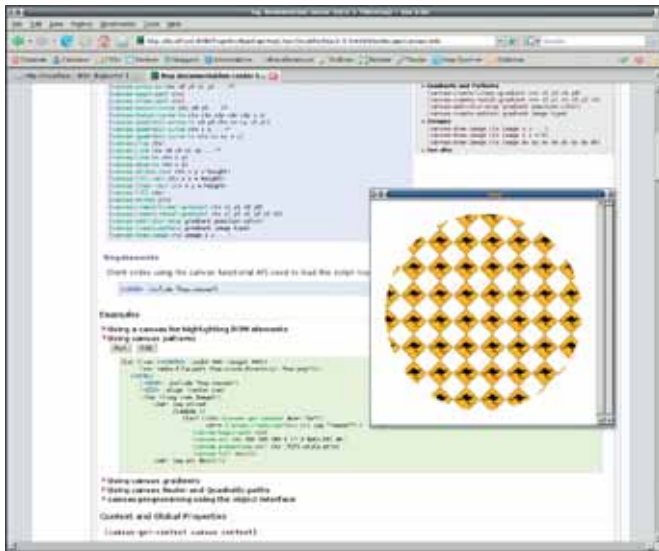
Fig. 1: Une galerie de photos sur le Web.

visible que lorsqu'il sera inclus dans un arbre HTML affiché par le client. Les dimensions de ce canevas ne peuvent pas être fixées par le courtier puisqu'il ne connaît pas les dimensions de la fenêtre mise à disposition par le client. Ainsi, pour ne déterminer les dimensions du canevas qu'au moment de son inclusion dans l'interface affichée par le client, les expressions fixant sa taille sont précédées du caractère ~ afin que ces expressions ne soient évaluées que lorsqu'elles atteindront le client.

La possibilité de spécifier où une expression doit être évaluée constitue une différence importante entre HOP et HTML. En HTML les attributs qui ne sont ni des URLs ni des renvois (ou callbacks) associés à des événements de l'interface graphique sont des littéraux exprimés sous forme de chaînes de caractères. En HOP, les attributs sont produits par des expressions du langage. Ainsi, dans notre exemple, les dimensions du canevas sont calculées, sur le client, à partir des dimensions de sa fenêtre (obtenues au moyen des deux fonctions `current-window-width` et `current-window-height`).

Notre application Web consiste principalement en un service HOP nommé `gallery` qui prend un unique argument (ici nommé `repo`) nommant, sur le courtier, le répertoire où trouver les photos à afficher. Ce service est défini par :

```
5: (define-service (gallery repo)
6:   (<HTML>
7:     (<HEAD> :include "hop-canvas"
8:       (<TITLE> "Une galerie de photos sur le Web")
9:       (<STYLE> "img.thumbnail { border: 2px solid transparent }"
10:        "img.thumbnail:hover { border-color: red }"))
11:     (<BODY> :style "background: black; color: white"
```



```

12:
13:   ~ (define (rotate-and-zoom-photo canvas img)
14:     (let ((ctx (canvas-get-context canvas "2d")))
15:       (canvas-properties-set! ctx
16:         :line-width 8
17:         :line-cap "round"
18:         :stroke-style "rgba(0,0,200,0.93)")
19:       (let loop ((angle 0)
20:                 (zoom 0.1))
21:         (canvas-clear-rect ctx 0 0 canvas.width canvas.height)
22:         (canvas-save ctx)
23:         (canvas-translate ctx (/ canvas.width 2) (/ canvas.
height 2))
24:         (canvas-rotate ctx angle)
25:         (canvas-scale ctx zoom zoom)
26:         (canvas-draw-image ctx img (- (/ img.width
27:                                           2)) (- (/ img.height 2))
28:                               img.width img.height)
29:         (canvas-restore ctx)
30:         (when (> (* 2 Math.PI) angle)
31:           (after 50
32:             (lambda ()
33:               (loop (min (* 2 Math.PI) (+ 0.4 angle))
34:                     (min 1.0 (+ 0.1 zoom))))))
35:         (<GALLERY> (or repo "/tmp/repo"))))

```

La similitude existant entre HTML et HOP est frappante sur cet exemple. En particulier, on peut voir que le document produit par le service possède un titre (ligne 8) et qu'il utilise une feuille de style (ligne 9 et 10). Par ailleurs, cet exemple illustre aussi une différence importante entre ces deux formalismes : en HOP un programme peut définir de nouvelles balises (ici GALLERY) pour étendre l'ensemble initialement fourni par le système, alors qu'en HTML l'ensemble des balises disponibles est figé. Afin de pouvoir utiliser les fonctions de dessin, un programme HOP doit contenir une déclaration qui provoque le chargement, dans le client, de

la bibliothèque d'exécution hop-canvas (ligne 7). Le code client peut alors utiliser les fonctions canvas-get-context (ligne 14), canvas-draw-image (ligne 26) ou encore canvas-stroke-rect (ligne 28). Toutes ces fonctions de dessins sont utilisées dans la fonction rotate-and-zoom-photo (ligne 13) qui prend en charge l'animation des photographies. Cette dernière constitue la partie la plus volumineuse de notre application.

La fonction client (puisque précédée du signe ~) rotate-and-zoom-photo prend en paramètre un canevas et une image qui est affichée d'abord très rétrécie puis de plus en plus grande. Pour produire cet effet, notre fonction utilise une boucle (ligne 19) qui est re-exécutée toutes les 50 millisecondes (ligne 31). À chaque pas de la boucle, l'angle de rotation de l'image est incrémenté ainsi que la taille d'affichage de l'image. La boucle s'arrête lorsque l'image a effectué une rotation complète (ligne 30). À chaque pas de la boucle, après avoir affiché l'image (ligne 26) notre programme lui ajoute un cadre (ligne 28). Les caractéristiques visuelles du cadre étant inchangées d'une itération à l'autre, elles sont fixées une fois pour toute avant d'entrer dans la boucle (ligne 15). La figure snapshot présente une copie d'écran de notre application exécutée dans Firefox (Fig.1).

La lecture des photographies sur le courtier

Afin de compléter notre application il ne nous reste plus qu'à spécifier les parties destinées à être exécutées par le courtier. Tout d'abord, étudions la définition de la nouvelle balise <GALLERY> qui a été utilisée précédemment :

```

37: (define (<GALLERY> dir)
38:   (<TABLE>
39:     <TR>
40:       (<TD> (map <THUMB> (directory->path-list dir)))
41:       (<TD> canvas)))

```

Comme on peut le constater, définir une nouvelle balise revient juste à définir une nouvelle fonction (ligne 37). Cette balise crée un tableau à deux colonnes. La première est destinée aux vignettes des photos, la seconde à l'animation des photos. La liste des photos (c'est-à-dire des fichiers) disponibles est produite par la fonction de la bibliothèque standard directory->path-list. Bien entendu, cette fonction accédant aux unités de stockage du courtier, elle ne peut être utilisée que par ce dernier.

```
42: (define (<THUMB> path)
43:   (<DIV>
44:     (<DIV> (exif-date (jpeg-exif path)))
45:     (<IMG> :class "thumbnail"
46:       :src ~($gallery/thumbnail $path)
47:       :onclick ~((let ((img (<IMG>)))
48:         (add-event-listener!
49:           img "load"
50:           (lambda () (rotate-and-zoom-photo $canvas img)))
51:         (set! img.src $path))))))
```

La balise <THUMB> crée les vignettes des photos. Pour cela, elle utilise les fonctions de la bibliothèque multimédia de HOP jpeg-exif et exif-date. La première lit un fichier de photo et extrait les informations EXIF qu'elle recèle. La seconde en extrait la date. Notre balise <THUMB> construit donc une boîte graphique dont la première ligne est constituée de la date où la photo a été prise. La seconde ligne est constituée de la vignette elle-même. Une action client est associée aux clics souris (ligne 48). Elle consistera à créer dynamiquement sur le client un nouvel objet contenant une image (ligne 51) dont le contenu sera fourni par le fichier de photo lui-même. Une fois la photo chargée par le client, elle sera transmise à la fonction précédemment étudiée rotate-and-zoom-photo qui la fait apparaître à l'écran du client.

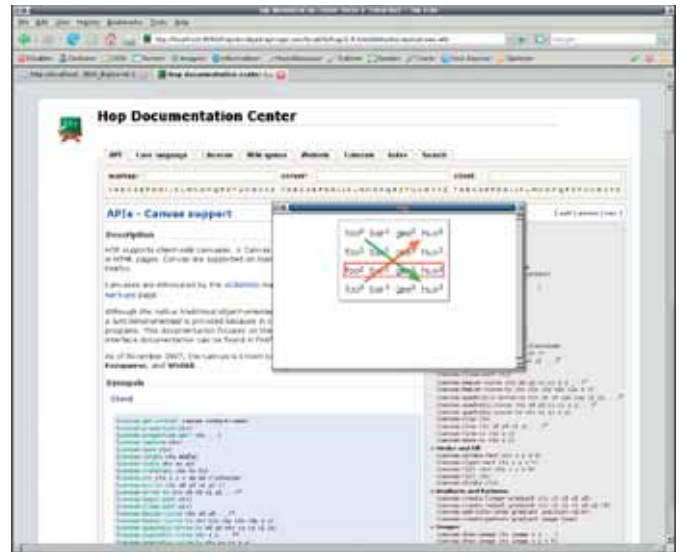
Le contenu de la vignette est donné par un nouveau service qui extrait le thumbnail de la zone de données EXIF au moyen de la fonction exif-thumbnail disponible dans la bibliothèque standard de HOP :

```
52: (define-service (gallery/thumbnail photo)
53:   (exif-thumbnail (jpeg-exif photo)))
```

Les types MIME des réponses HTTP sont inférés automatiquement par HOP à partir des valeurs servies. Cela simplifie encore un peu plus l'écriture des programmes qui n'ont ainsi pas besoin de calculer explicitement ces types.

L'implantation que nous avons proposée ci-dessus est compatible avec tous les navigateurs Web. Toutefois, une version plus simple peut-être utilisée pour les navigateurs qui supportent les images dites inline (c'est-à-dire tous les navigateurs sauf Internet Explorer). En effet, pour ces navigateurs, il n'est pas utile de créer un service pour produire le contenu des vignettes car ce dernier peut-être directement envoyé au client avec la vignette elle-même. Ainsi, en utilisant des images inline dans la balise <THUMB>, le service gallery/thumbnail devient inutile :

```
42: (define (<THUMB> path)
43:   (<DIV>
44:     (<DIV> (exif-date (jpeg-exif path)))
```



```
45:   (<IMG> :class "thumbnail"
46:     :src ~(string-append "data:image/jpeg;base64,"
47:       (base64-encode (exif-thumbnail exif)))
48:     :onclick ~((let ((img (<IMG>)))
49:       (add-event-listener!
50:         img "load"
51:         (lambda () (rotate-and-zoom-photo $canvas img)))
52:       (set! img.src $path))))))
```

Conclusion

Cet article montre la simplicité de développer des applications multimédia sur le Web avec HOP. Pour s'en convaincre il suffit de comparer l'exemple de cet article à ceux présentés dans de précédents articles de Programmez! (dans le N° 99) qui ont montré des applications similaires, par exemple écrites avec Silverlight. Alors que dans les précédents articles il fallait utiliser plusieurs syntaxes, plusieurs formalismes et des programmes assez volumineux, HOP permet, en une cinquantaine de lignes de code et en n'utilisant qu'un formalisme unique, de développer une application complète et facile à déployer.

Une version HOP, de cet article est disponible à l'adresse : <http://hop.inria.fr/hop/weblots>. Une fois HOP installé sur votre machine, cette weblet, vous permettra d'exécuter et modifier à votre guise, le code présenté dans cet article.

■ Manuel Serrano

Directeur de Recherche, Inria
www.inria.fr/mimosa/Manuel.Serrano

■ Christian Queinnec

Professeur, Université Pierre et Marie Curie
www.spi.lip6.fr/~queinnec/

L'information permanente

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

www.programmez.com

RIA pour téléphones mobiles

2^e partie

Les applications internet riches pour les téléphones mobiles se heurtent à des contraintes de portabilité largement plus grandes que celles du Web. J2ME propose des interfaces pour construire ces applications qui possèdent l'avantage d'être basées sur une standardisation et d'être disponibles sur une large gamme de téléphones. Cette seconde partie détaille la mise en oeuvre sous J2ME de la communication entre le téléphone mobile et le serveur d'application.



L'interaction avec l'utilisateur ne représente que la partie émergée de l'application riche, la communication avec le serveur en est la partie en arrière plan. Cette communication utilise les protocoles de référence du Web : HTTP et XML, mais la contingence des ressources au niveau des téléphones nécessite une attention particulière. J2ME possède un framework de gestion des entrées-sorties qui permet d'interagir avec les systèmes extérieurs que ce soit le système de fichier, le bluetooth ou la connectivité IP.

L'exemple 3 présente la classe **HttpRequest** pour effectuer une requête GET et récupérer une réponse. L'initialisation de la requête se fait avec le framework générique en utilisant la classe **Connector** :

```
hc = (HttpConnection)Connector.open(url, Connector.READ_WRITE);
hc.setRequestMethod(HttpConnection.GET);
```

HttpConnection représente la connexion HTTP, **setRequestMethod** permet de positionner les commandes les plus courantes (GET, POST), il est aussi possible de positionner les paramètres d'en-tête. Le déclenchement de la requête HTTP se fait lors de l'appel pour lire la réponse :

```
rc = hc.getResponseCode();
```

La récupération de la réponse par l'intermédiaire de **openInputStream** qui fournit un **InputStream** pour accéder au un flux de données.

Il est à noter que si l'interface **HttpConnection** permet de couvrir un large éventail de fonctionnalités, les implémentations sont, elles, souvent assez limitées. Par exemple, sur certains mobiles, les requêtes POST utilisent exclusivement un mode " chunked " qui n'est pas supporté par tous les serveurs " proxy ".

Ces limitations conduisent à préférer des protocoles optimisés comme REST plutôt que le protocole SOAP pour la communication applicative. L'interface fournie est dépouillée d'outils comme l'encodage MIME qui permet de transmettre sur HTTP les éléments multimédia. Pour une utilisation pratique de la connectivité HTTP, il est nécessaire de développer une bibliothèque d'utilitaires " bas niveau " pour les fonctions de manipulation des données HTTP.

Le parsing XML est une fonctionnalité également absente du profil J2ME standard. Mais une excellente librairie applicative: KXML, offrant le parsing XML, est disponible pour J2ME. KXML permet de réaliser le parsing XML en respectant les deux règles principales pour une librairie J2ME : utiliser peu de ressources et être portable.

Le parsing XML sous J2SE est implémenté par deux interfaces :

- SAX permettant de manipuler le flux au niveau des tags
- DOM représentant le XML comme un document

KXML présente une approche mixte entre SAX et DOM. KXML implémente une interface **XmlPullParser** qui est assez semblable à l'interface SAX avec la différence de nécessiter un appel explicite pour traiter chaque élément XML. KXML offre un package **kdom** pour construire des documents au-dessus de ce parseur.

KXML possède aussi la capacité de parser le WBXML qui est la norme XML compressée dont l'utilisation est recommandée pour la communication d'éléments XML sur GPRS.

L'exemple 4 présente une utilisation de KXML pour une application simplifiée de lecture de flux RSS. Le format RSS est extrêmement simple et n'utilise ni attributs, ni namespaces différents :

```
<rss>
<channel>
<title> ... </title>
<description>...</description>
<item>
<title> ... </title>
<description>...</description>
</item>
...
<item>
<title> ... </title>
<description>...</description>
</item>
</channel>
</rss>
```

Le parseur est instancié par une factory qui permet de le construire avec les bonnes fonctionnalités : pas de validation et de gestion de namespaces dans ce cas.

```
factory.setFeature(XmlPullParser.FEATURE_PROCESS_NAMESPACES, false);
factory.setFeature(XmlPullParser.FEATURE_VALIDATION, false);
```

La boucle principale récupère un événement par l'appel à la méthode **getEventType** de :

- Début de document, fin de document
- Début de tag, Fin de tag
- Texte

Le parser possède les éléments de contextes associés à l'événement : namespace et attributs. Deux parseurs sont utilisés dans cet exemple, un pour le flux RSS, l'autre pour purger les éventuels tags HTML présents dans le titre ou la description. Pour le nettoyeur de HTML il est

nécessaire de développer une classe **StringReader** (un des manques de J2ME).

Une autre approche pour réaliser le parsing HTML est d'utiliser une extension de J2ME : le JSR172 qui comprend un parser XML. Cette fonctionnalité est pour l'instant, comme toutes les fonctionnalités avancées, limitée à un nombre restreint de terminaux.

Un champ d'applications à conquérir

Pour une cible d'applications riches visant le plus large éventail de téléphones mobiles, J2ME possède de nombreux atouts: sa diffusion sur les téléphones mobiles, une garantie d'un bon niveau de compatibilité entre équipements, un fonctionnel qui s'enrichit régulièrement (interfaces 3D, GPS...) et une population de développeurs formés en environnement Java.

Le développement en Java pour Mobile nécessite une attention particulière à la consommation de ressources, la portabilité entre équipements et aux tests sur différents mobiles.

Une fois ces facteurs pris en compte il se révèle très proche de n'importe quel autre développement Java : un seul langage, même outils (Netbeans, Eclipse), même conception " OO " et des composants logiciels très semblables : Listener, Graphics, classes de base...

■ Olivier Théry

```
package exemple3;

import java.io.*;
import java.util.Enumeration;
import java.util.Hashtable;
import java.util.Vector;
import javax.microedition.io.*;

public class HTTPRequest {
    private Hashtable headers;
    private int returnCode;
    private InputStream is;
    private String contentType;
    private int contentLength;

    public boolean doGet(String url, Hashtable h) {
        HttpURLConnection hc = null;
        is = null;
        headers = null;
        contentType = null;
        contentLength = -1;
        returnCode = -1;

        try {
            for(int count = 0; count < 3; count++) { // On fait au maximum
                3 redirections
                hc = (HttpURLConnection)Connector.open(url, Connector.READ_WRITE);
                hc.setRequestMethod(HttpURLConnection.GET);
                if(h != null) { // Positionne les entetes optionnels
```

```
Enumeration en = headers.keys();
while(en.hasMoreElements()) {
    String s = (String)en.nextElement();
    hc.setRequestProperty(s, (String)h.get(s));
}
}
returnCode = hc.getResponseCode();
if((getReturnCode() != 301 && getReturnCode() != 302)) //
verification du redirect
    break;
url = hc.getHeaderField("Location");
hc.close();
}
is = hc.openInputStream();
contentLength = hc.getHeaderFieldInt("content-length", -1);
contentType = hc.getHeaderField("Content-Type");
headers = new Hashtable();
for(int i = 0; hc.getHeaderFieldKey(i) != null; i++) {
    headers.put(hc.getHeaderFieldKey(i).toLowerCase(), hc.getHeader
Field(i));
}
hc.close();
} catch (IOException ex) {
    ex.printStackTrace();
    return false;
}

return true; // success
};
}
```

Exemple 3 – Gestion d'une requête http

```
package exemple4;

import java.io.Reader;
import java.util.Vector;
import org.xmlpull.v1.XmlPullParser;
import org.xmlpull.v1.XmlPullParserFactory;

public class RssFeed {
    private Vector items;
    private String title;
    private String description;
    private String image;

    public RssFeed() {
        items = new Vector();
    }

    public void parse(Reader inputReader, boolean utf8) {
        XmlPullParserFactory factory;
        XmlPullParser xpp;
```

```

String tag;
RssItem rss = null;
String text = "";

try {
    factory = XmlPullParserFactory.newInstance("org.xml2.io.K
XmlParser", null);

    factory.setFeature(XmlPullParser.FEATURE_PROCESS_NAMESPACES, false);
    factory.setFeature(XmlPullParser.FEATURE_VALIDATION, false);

    xpp = factory.newPullParser();

    xpp.setInput(inputReader);
    xpp.defineEntityReplacementText("acirc", "à");
    xpp.defineEntityReplacementText("agrave", "á");
    xpp.defineEntityReplacementText("ocirc", "ó");
    xpp.defineEntityReplacementText("ucirc", "ü");
    xpp.defineEntityReplacementText("ugrave", "ù");
    xpp.defineEntityReplacementText("icirc", "í");
    xpp.defineEntityReplacementText("eacute", "é");
    xpp.defineEntityReplacementText("ecirc", "ê");
    xpp.defineEntityReplacementText("egrave", "è");
    xpp.defineEntityReplacementText("ccedil", "ç");
    int eventType = xpp.getEventType();

    do {
        if(eventType == XmlPullParser.START_DOCUMENT) {
        } else if(eventType == XmlPullParser.END_DOCUMENT) {
            break;
        } else if(eventType == XmlPullParser.START_TAG) {
            tag = xpp.getName();
            if(tag.equals("item")) {
                rss = new RssItem();
                text = "";
            }
        } else if(eventType == XmlPullParser.END_TAG) {
            if(xpp.getName().equals("item")) {
                if(rss.title != null)
                    items.addElement(rss);
                rss = null;
            } else if (xpp.getName().equals("title")) {
                if(rss != null)
                    rss.title = PurgeHTML.clean(text).trim();
                else
                    title = PurgeHTML.clean(text).trim();
            } else if (xpp.getName().equals("description")) {
                if(rss != null)
                    rss.description = PurgeHTML.clean(text).trim();
                else
                    description = PurgeHTML.clean(text).trim();
            } else if (xpp.getName().equals("image")) {
                image = text.trim();
            }
        }
        text = "";
    }

```

```

    } else if(eventType == XmlPullParser.TEXT) {
        int[] sl = new int[2];
        char c[] = xpp.getTextCharacters(sl);
        String s;
        if(utf8)
            s = cleanUTF8(c, sl[0], sl[1]);
        else
            s = new String(c, sl[0], sl[1]);
        text += s;
    }
    eventType = xpp.next();
} while (eventType != XmlPullParser.END_DOCUMENT);
return ;
} catch (Exception ex) {
    ex.printStackTrace();
}
}

public String cleanUTF8(char[] c, int s, int l) {

    char[]o = new char[c.length];
    int j = s;
    for(int i = s; i < s + l; i++) {
        // le seul cas de mapping interessant : facile entre UTF8 et
        ISO8859-1
        if((i < c.length - 1) && ((c[i] & 0xe0) == 0xc0) && ((c[i+1] &
        0xc0) == 0x80)){
            o[j++] = (char)(((c[i] & 0x3) << 6) | (c[i+1] & 0x3f));
            i++;
        } else if((c[i] & 0xc0) == 0xc0) {
            o[j++] = '*'; // * caractère inconnu
        } else if((c[i] & 0xc0) != 0x80) {
            o[j++] = c[i]; // * caractère pas purgé
        }
    }
    return new String(o, s, l);
}

public Vector getItems() {
    return items;
}

public String getTitle() {
    return title;
}

public String getDescription() {
    return description;
}

public String getImage() {
    return image;
}
}

```

Exemple 4

Démarrer avec ejabberd

ejabberd est un serveur de présence et de messagerie instantanée qui repose sur le protocole XMPP, également connu sous le nom de Jabber. C'est cette technologie qu'a choisie Google pour son service Google Talk et que l'on retrouve également dans sa plate-forme mobile Android. Il est distribué sous licence GNU GPL, est édité par Process-one (1) et est multi-plate-forme.

Ejabberd est écrit en Erlang (2). Il s'agit d'un langage fonctionnel Open Source développé par Ericsson, créé à l'origine pour être utilisé dans les dispositifs réseaux. Il est actuellement utilisé pour faire fonctionner des équipements GPRS, accélérateurs SSL, répartiteurs de charge (certains boîtiers Althéon) ou switchs ATM, dont le fameux AXD301 d'Ericsson, qui peut se targuer de fournir une disponibilité de neuf neufs (99.9999999%, soit 31 ms d'indisponibilité par an...) (3). Erlang a depuis évolué vers un langage plus généraliste, mais nombre de ses spécificités résultent de son background réseau. En effet, les équipements réseaux doivent pouvoir supporter un grand nombre de connexions, ce qui implique de pouvoir gérer de nombreux threads. Erlang est effectivement *thread oriented* : les threads Erlang sont très légers et très simples à manipuler. Une demi-seconde suffit pour créer facilement 100 000 threads sur un PC de base, chaque thread occupant un minimum de mémoire.

Ensuite, comme on l'a vu, le langage doit permettre de limiter au maximum les indisponibilités, même en cas d'erreurs dans le programme. Erlang fournit pour cela un système d'arbre de supervision qui permet de relancer automatiquement une tâche qui se "planterait", sans incidence sur le fonctionnement général de l'application. Il est également possible de mettre à jour le code de l'application à chaud, sans interruption de service. Enfin, pour augmenter encore la disponibilité et la performance, les applications doivent pouvoir tourner sur plusieurs nœuds en parallèle. C'est pourquoi il est possible avec Erlang de manipuler les threads distants (c'est-à-dire tournant sur une autre machine physique) de la même manière que les threads locaux, de façon totalement transparente. Des mécanismes de réplication sont également fournis pour partager les données entre les nœuds. Il en résulte que les applications écrites en Erlang peuvent être clusterisées de façon pratiquement automatique. Pour résumer, les contraintes qui ont présidé à la naissance d'Erlang en ont fait un langage concurrent, fiable et réparti, et l'on va retrouver naturellement ces caractéristiques dans ejabberd. Des clusters ejabberd gèrent actuellement en production plusieurs centaines de milliers d'utilisateurs en simultané pour des sociétés dont c'est le cœur de métier.

Installation

ejabberd est fourni nativement ou sous forme de paquetages additionnels pour les systèmes d'exploitation suivants : Debian, Fedora Cora, Gentoo, Mandriva, OpenSuse, Ubuntu, FreeBSD port, Nexenta, Gnu/OpenSolaris. Si vous désirez l'installer sur un autre système d'exploitation ou si la version disponible est trop ancienne, des installateurs graphiques sont également disponibles (4) pour Linux, Mac OS X et MS-Windows. Ces installateurs incluent Erlang et ejabberd. En quelques clics,

vous pouvez disposer d'un ejabberd installé, configuré et démarré. L'application est également fournie sous forme de sources. Dans ce cas, un classique `./configure && make && sudo make install` permet de l'installer. Vous devez disposer des bibliothèques de développement *iconv*, *expat*, *zlib*, *pam* et *openssl* et avoir installé Erlang préalablement. Pour tester la dernière version de développement, tournez-vous vers le dépôt Subversion (5). ejabberd repose sur une ou plusieurs bases de données pour l'authentification (Jabber ID, mots de passe), les profils utilisateurs (vCards), les listes de contacts (rosters) ou le stockage des messages différés (offline messages). Dans l'installation par défaut, la base de données interne à Erlang (Mnesia) est utilisée, ce qui ne requiert aucune installation ou configuration supplémentaire. Mais il est également possible d'utiliser LDAP, PostgreSQL, MySQL ou un connecteur ODBC pour d'autres bases (Oracle, Firebird, SQL Server, etc.) pour stocker ces données. Un mécanisme d'authentification externe est également disponible pour gérer l'authentification via un programme tiers.

Fonctionnalités

Une fois installé, ejabberd offre les fonctionnalités suivantes, en plus des classiques gestion de présence et messagerie instantanée :

- le Multi-User Chat (MUC), fournit des salons de discussions paramétrables ;
- le Jabber User Directory permet de rechercher dans les vCards des utilisateurs enregistrés ;
- les Privacy Lists, ou listes de blocage, permettent de gérer finement sa présence ou d'interdire tout contact par d'autres utilisateurs ou groupe d'utilisateurs, pour ne pas être dérangé ;
- les Shared Rosters, ou listes de contacts partagées, permettent de pré-définir des listes de contacts ;
- le HTTP-Polling et le HTTP-Binding permettent d'encapsuler du XMPP dans du HTTP, pour traverser des pare-feux ou utiliser un client Web, par exemple ;
- le Publish and Subscribe (PubSub) permet de publier et de recevoir de l'information en temps réel. Ce mécanisme a par exemple été utilisé lors de la dernière coupe du monde de football en Allemagne pour publier en temps réel les résultats des matchs ;
- le Virtual Hosting autorise la gestion de plusieurs domaines sur un même serveur ;
- le support des passerelles (transports) permet de communiquer avec des systèmes de messagerie tiers. Des passerelles XMPP vers AIM, ICQ, MSN, Yahoo Messenger ou SMS sont disponibles. ejabberd est livré avec un transport IRC ;
- l'utilisation de SASL et de TLS/STARTLS garantit des communications sécurisées.

(1) <http://www.process-one.net/>

(2) <http://www.erlang.org/>

(3) <http://www.pragmaticprogrammer.com/articles/erlang.html>

(4) <http://www.process-one.net/en/ejabberd/downloads>

(5) <http://svn.process-one.net/ejabberd/trunk>

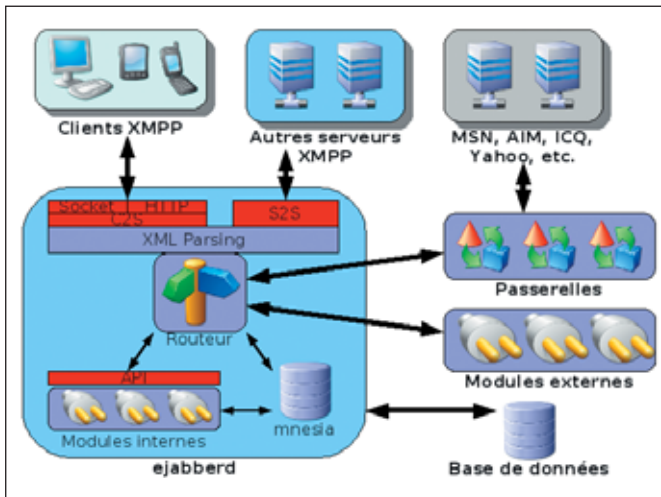


Figure 1 : architecture d'un nœud ejabberd

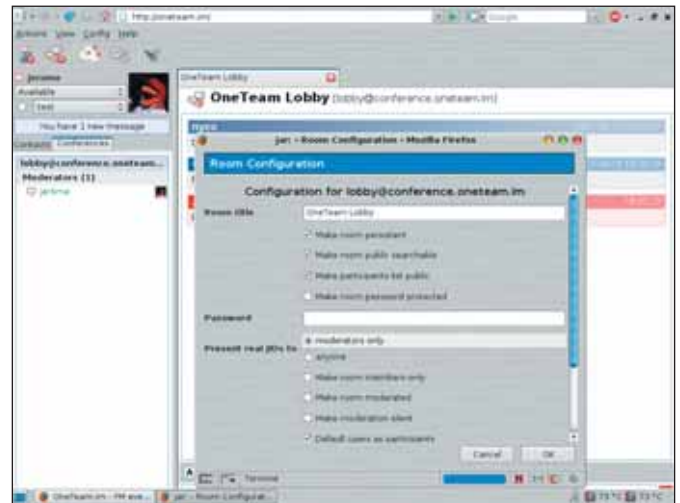
Architecture

La figure 1 représente un nœud ejabberd et ses relations avec le monde extérieur. Suivons le parcours d'un paquet XMPP pour comprendre comment fonctionne ejabberd, dans le cas d'un simple envoi de message. Le client XMPP de l'expéditeur du message est connecté sur une connexion C2S : Client To Server. La plupart du temps, il s'agit d'une connexion TCP sur le port XMPP (5222) mais il est également possible d'être connecté en HTTP (dans ce cas, on encapsule le XMPP dans des paquets HTTP). L'expéditeur envoie donc un message sur sa connexion C2S, sous la forme d'un paquet XMPP qui ressemble à :

```
<message
to='romeo@example.net'
from='juliet@example.com/balcony'
type='chat'>
  <body>Wherefore art thou, Romeo?</body>
</message>
```

Ici, romeo@example.net est l'identifiant Jabber (JID) du destinataire. juliet@example.com est le JID de l'expéditeur.

XMPP est un protocole basé sur XML, la première tâche pour le serveur est donc de parser le paquet XMPP pour le transformer dans une structure de donnée interne. Le paquet est ensuite injecté dans le routeur, qui est le cœur d'ejabberd. Le routeur va consulter la table de routage, qui contient l'ensemble des utilisateurs actuellement connectés au serveur. Cette table est stockée dans la base de données interne de ejabberd, appelée Mnesia (ce SGBD est fourni en standard avec Erlang). Si le JID du destinataire est présent dans la table de routage, le routeur y récupère l'identifiant de sa connexion C2S, et y envoie le paquet XMPP. Si en revanche le destinataire n'est pas actuellement connecté, le routeur va d'abord vérifier qu'il existe, en consultant la liste des utilisateurs enregistrés. Cette dernière peut être stockée dans la base Mnesia, dans une base relationnelle externe ou dans un annuaire LDAP. Si le destinataire est trouvé, alors le routeur envoie le message à un module interne dont le rôle est de gérer les messages différés (offline messages). Ce module alors va stocker le paquet dans une base de données (Mnesia ou un SGBD/R). Lorsque le destinataire se connecte enfin, le routeur va prévenir le module qu'un utilisateur vient de s'au-



Configuration d'un salon de discussion en utilisant le client OneTeam (<http://oneteam.im/>) ;

thentifier. Le module va vérifier dans la base si des messages sont en attente et les ré-injecter dans le routeur le cas échéant. Comme précédemment, le routeur va consulter la table de routage, récupérer la connexion C2S du destinataire et lui envoyer le message.

Il existe également un troisième cas de figure. En effet, XMPP est un protocole qui repose sur un réseau de serveurs et non pas sur un gros serveur centralisé comme c'est par exemple le cas pour MSN ou Yahoo messenger. L'expéditeur peut donc être enregistré sur un autre serveur XMPP n'importe où dans le monde. Le domaine XMPP correspond à la partie droite du JID (exemple.net dans notre exemple) et identifie le serveur associé à un utilisateur.

Si le domaine ne correspond à aucun domaine géré par le serveur local, alors il s'agit d'un utilisateur enregistré sur un autre serveur. Dans ce cas, ejabberd va initier une connexion S2S (Server To Server). Il s'agit d'une connexion TCP vers le port 5269 sur le serveur distant (ejabberd peut également effectuer une requête DNS SRV pour obtenir le numéro de port s'il n'est pas standard). Le paquet est ensuite envoyé sur cette connexion. De manière symétrique, ejabberd peut recevoir des paquets en provenance d'autres serveurs via des connexions S2S entrantes. Dans ce cas, le paquet XML est parsé, injecté dans le routeur et traité de la même façon que pour un paquet provenant d'une connexion C2S. Enfin, le mécanisme de transport permet d'interfacer un serveur XMPP avec des systèmes de messagerie instantanée propriétaires tiers. Si par exemple le destinataire du message possède un JID qui ressemble à romeo@msn.example.net, alors le routeur va envoyer le paquet à msn.example.net qui est le nom du transport MSN. Ce transport peut être un composant interne à ejabberd (comme le module de gestion des messages différés) ou un composant externe. Dans ce dernier cas, ce composant est connecté à ejabberd via une connexion TCP (Jabber Component Protocol, il s'agit du troisième et dernier type de connexion XMPP, avec C2S et S2S). Le routeur de ejabberd envoie le paquet sur cette connexion, qui est traduit par le transport depuis XMPP vers le format du protocole propriétaire et envoyé vers le serveur distant (MSN, Yahoo, AIM ou autre).

Développement de modules

Nous avons vu que ejabberd se repose sur un module interne pour gérer les messages différés. En fait, mis à part le cœur de l'application (le

routeur), l'ensemble des fonctionnalités de l'application (l'authentification, la gestion des rosters, les vCards, le PubSub, etc.) sont prises en charge par des modules. Cette architecture modulaire permet de personnaliser le fonctionnement du serveur sans avoir à modifier l'application elle-même. Comme nous allons le voir dans l'exemple suivant, ejabberd propose une API permettant de développer très facilement de nouveaux modules. Regardons comment fonctionne le module `mod_time`, fourni en standard avec ejabberd et qui permet d'envoyer une requête au serveur pour connaître son heure. Par exemple, si vous envoyez la requête suivante au serveur :

```
<iq to="example.net" type="get">
  <query xmlns="jabber:iq:time"/>
</iq>
```

Le serveur répondra par :

```
<iq from='example.net' to='jerome@example.net/OneTeam' type='result'>
  <query xmlns='jabber:iq:time'>
    <utc>20071128T13:43:24</utc>
  </query>
</iq>
```

Une requête IQ (Info Query) de type *get* permet de demander une information au serveur qui répondra par un IQ de type *result*. La sémantique de la requête est spécifiée par le namespace du query, ici, `jabber:iq:time`. Pour information, ce namespace et le format de la requête sont décrits dans la spécification XEP-0090 (7). Voyons comment nous pouvons écrire un module pour gérer cette requête. Un module ejabberd est un module Erlang qui doit exporter au moins deux fonctions dans son interface : *start* et *stop*. La fonction *start* est appelée lors du démarrage du module. Ici, elle enregistre un handler auprès du routeur qui recevra toutes les requêtes IQ avec le namespace `jabber:iq:time` :

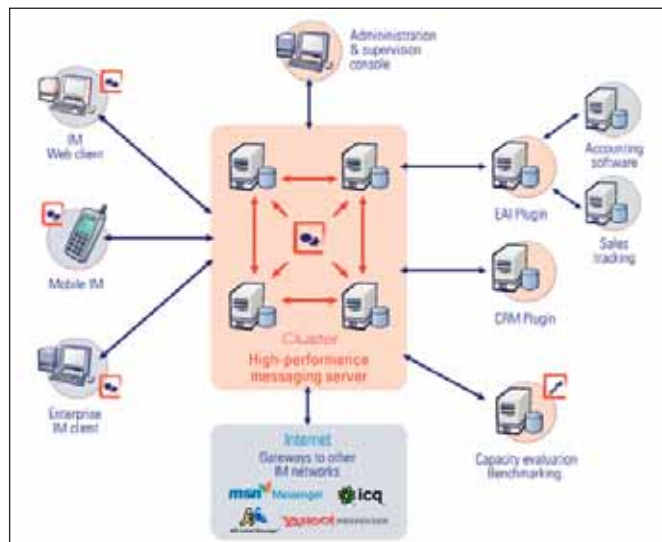
```
start(Host, Opts) ->
  gen_iq_handler:add_iq_handler(ejabberd_local, Host, "jabber:iq:time",
    ?MODULE, process_local_iq, one_queue).
```

Les deux paramètres de *start* sont le domaine XMPP pour lequel le module est démarré (un serveur ejabberd peut gérer plusieurs domaines sous forme de virtual hosts) et la liste de paramètres du module en provenance du fichier de configuration du serveur (nous pouvons l'ignorer, ce module n'a pas de paramètres).

Ensuite, nous appelons la fonction `add_iq_handler` du module `gen_iq_handler` qui est une fonction de l'API qui permet d'enregistrer le handler qui va gérer les IQ. Les paramètres de cette fonction sont :

- `ejabberd_local` : demande à recevoir les IQ envoyés au serveur (il est également possible de recevoir des IQ envoyés aux utilisateurs) ;
- `Host` : Virtual Host pour lequel le module veut recevoir les IQ ;
- `"jabber:iq:time"` : namespace pour lequel on veut s'enregistrer ;
- `?MODULE` et `process_local_iq` indique le handler, c'est-à-dire la fonction à appeler pour traiter les IQ reçus (ici, la fonction `process_local_iq` du module courant) ;
- Le dernier paramètre permet de préciser comment traiter plusieurs IQ reçus simultanément : en parallèle, en les plaçant dans une queue FIFO, etc. L'argument `one_queue` utilise une queue FIFO.

Il faut donc maintenant écrire le handler qui va traiter les IQ, à savoir la fonction `process_local_iq` :



```
process_local_iq(From, To, #iq{type = Type, sub_el = SubEl} = IQ) ->
  case Type of
  set ->
    IQ#iq{type = error, sub_el = [SubEl, ?ERR_NOT_ALLOWED]};
  get ->
    UTC = jlib:timestamp_to_iso(calendar:universal_time()),
    IQ#iq{type = result,
      sub_el = [{xmlelement, "query", [{"xmlns", "jabber:iq:time"}],
        [{xmlelement, "utc", [],
          [{xmldata, UTC}]}]}]}
  end.
```

Le handler reçoit en premier et deuxième paramètre l'expéditeur et le destinataire de l'IQ. Le troisième paramètre est le paquet IQ lui-même, dans une structure de donnée Erlang. Dès la déclaration de ce troisième argument, on récupère le type de l'IQ (l'attribut `type` sur le tag `<iq>`) et on vérifie qu'il s'agit bien d'un *get* grâce à l'instruction `case`. Si c'est un *set*, on renvoie un IQ de type *error* (il n'est pas possible de changer l'heure du serveur avec un IQ...).

Si c'est bien un *get*, on récupère l'heure courante dans la variable `UTC` et on renvoie l'IQ de type *result* en y ajoutant comme sous-élément le `<query>` avec le bon namespace et contenant le `<utc>` attendu.

Il ne reste plus qu'à écrire la fonction *stop*, qui va désinscrire le handler, pour ne plus qu'il reçoive les IQ :

```
stop(Host) ->
  gen_iq_handler:remove_iq_handler(ejabberd_local, Host, "jabber:iq:time").
```

En quelques lignes, nous avons écrit un module ejabberd, certes simple, mais totalement fonctionnel. Le traitement des IQ n'est qu'un aspect du protocole XMPP et ejabberd dispose bien sûr d'une API complète (8) permettant de gérer la plupart des autres opérations qu'un serveur Jabber peut être amené à traiter.

■ **Jérôme Sautret** - Directeur technique de Process-one.

(7) <http://www.xmpp.org/extensions/xep-0090.html>

(8) http://www.process-one.net/en/wiki/ejabberd_module_development/

Maîtriser l'API Preferences de Java

La persistance d'informations relatives à la configuration est une fonctionnalité de base des applications modernes. Java apporte une réponse par son API Preferences, dont nous découvrons le maniement et les subtilités.

Il est de nos jours habituel de voir une application ouvrir une fenêtre à la taille que celle-ci avait lors de la précédente session. On n'imaginerait pas non plus devoir saisir les informations d'un compte à chaque utilisation d'un logiciel de messagerie. Ces petites informations, qui ne justifient en général pas l'utilisation d'une base de données, sont normalement sauvegardées dans la base de registre sous Windows, ou dans des fichiers de configuration sous des systèmes de type Unix. Un langage comme Java doit fournir une API encapsulant tout cela, de manière à ce que les applications se comportent à l'identique sur n'importe quel système hôte. Cette API s'appelle l'API Preferences et ses classes résident dans le paquetage `java.util.prefs`. Très simple d'utilisation elle ne parvient que moyennement à atteindre son but, ce "moyennement" se révélant a priori suffisant en ce qui concerne les besoins courants.

1 Des structures de données arborescentes

Il est naturel d'employer des arborescences pour organiser les données de configuration que nous devons manipuler. Ainsi, une application peut vouloir sauvegarder des informations de base qui seront à la racine de l'arborescence et des informations relatives à l'interface utilisateur qui seront rangées sous un nœud 'gui' par exemple. La base de registre Windows, base de données hiérarchique, fournit un environnement tout prêt, environnement qui sera simulé sous Unix/Linux au moyen d'une arborescence de répertoires et de fichiers au format XML.

2 Deux arborescences de données

Les concepteurs de l'API ont jugé bon de reprendre le schéma Windows dont la base de registre permet de contenir d'une part des données utilisateur à l'usage de l'utilisateur courant et seulement lui, et d'autre part des données dites systèmes qui seront communes à tous les utilisateurs. Il est entendu que utilisateur signifie utilisateur du système, et pas utilisateur de l'application. Ainsi, un logiciel de messagerie peut être lancé sous le compte "fred" d'un Linux. Dans ce cas l'utilisateur est fred, même si l'utilisateur réel du logiciel est une autre personne qui aurait configuré son compte dans le dit logiciel. En ce qui concerne l'arborescence système, le principe fonctionne très bien en lecture, mais on se doute que l'on va se heurter à des problèmes de droits d'accès en écriture.

3 Un premier exemple

L'API Preferences est très simple à manipuler. La démarche consiste à obtenir une instance de la classe `Preferences` via l'une de ses méthodes statiques. Les méthodes `userRoot` et `systemRoot` créent une instance "placée" à la racine de l'une ou l'autre des deux arborescences, tandis que les méthodes `userNodeForPackage` et `systemNodeForPackage`, permettent de se positionner directement sur un nœud, les nœuds manquants étant automatiquement créés le cas échéant. Une fois une instance obtenue, il est possible de naviguer dans l'arbores-

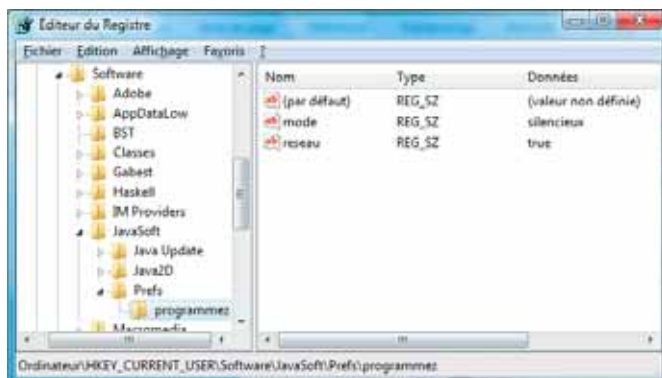


Fig. 1

cence au moyen de la méthode non statique `node`. Les méthodes suffixées `ForPackage` créent automatiquement une arborescence basée sur les paquetages de la classe. Le procédé présente l'avantage de la simplicité. Il est toutefois possible de créer une arborescence à la main. Voici un exemple :

```
package programmez;

import java.util.prefs.*;

public class MonApp {

    private void init() {
        String mode;
        boolean reseau;

        Preferences pref =
            Preferences.userNodeForPackage(MonApp.class);
        mode = pref.get("mode", "verbeux");
        reseau = pref.getBoolean("reseau", false);
        System.out.println("mode: " + mode);
        System.out.println("reseau: " + reseau);
    }

    private void quit() {
        Preferences pref =
            Preferences.userNodeForPackage(MonApp.class);
        pref.put("mode", "silencieux");
        pref.putBoolean("reseau", true);
    }

    public static void main(String[] args) {
        MonApp app = new MonApp();
        app.init();
        app.quit();
    }
}
```

```
app.init();
}
}
```

La classe `MonApp` fait partie du paquetage `programmez`, donc l'invocation `Preferences.userNodeForPackage(MonApp.class)` va créer un nœud `programmez` sous la racine courante. Celle-ci est celle qui aurait été retournée par l'invocation `Preferences.userRoot()`. Le reste du code parle de lui-même. La méthode `init` lit des valeurs, en fournissant, c'est obligatoire, des valeurs par défaut dans le cas où une clé n'existe pas encore. La méthode `quit` écrit des valeurs. Le résultat est illustré Fig. 1. Faisons deux remarques. D'abord la ligne de code :

```
Preferences.userNodeForPackage(programmez.MonApp.class);
```

Serait rigoureusement équivalente à celle de l'exemple. Ensuite, si l'on veut travailler manuellement avec l'arborescence, le code devient (code complet sur le Cd-Rom ou le site) :

```
Preferences pref = Preferences.userRoot();
pref = pref.node("/programmez");
```

Lorsque l'on travaille de cette manière, il importe d'être bien conscient que la chaîne de caractères que l'on passe à la méthode doit être formée comme une arborescence de répertoire Unix. Ainsi :

```
/programmez
```

Signifie à la racine telle que celle-ci aurait été retournée par `userRoot`, `root` signifiant ici racine du système de fichier, et non compte super-utilisateur, tandis que :

```
programmez
```

Signifie le nœud `programmez`, à partir du nœud sur lequel la classe `Preferences` est positionnée. Dans notre exemple, c'est le même endroit dans les deux cas, mais il n'en est pas obligatoirement toujours ainsi.

4 Emplacement des données et conséquences sur le fonctionnement

Ce que nous allons dire dans ce paragraphe a été vérifié pour une JDK 1.6.0._02. Cependant il est vraisemblable que cela s'applique à de nombreuses JVM sinon exactement sur la forme, du moins sur le fond. Regardons où sont sauvegardées nos données et ce que cela implique.

- Sous Linux et arborescence utilisateur :

Dans ce mode et en supposant que l'utilisateur s'appelle `fred`, les préférences sont enregistrées sous `/home/fred/.java/.userPrefs`. Le répertoire `.userPrefs` est la racine, et si l'on reprend notre exemple, un sous-répertoire `programmez` y serait créé et dans ce répertoire un fichier `prefs.xml` contiendrait les clés et leurs valeurs. Dans ce mode, il est parfaitement clair que le mécanisme fonctionne comme prévu et donne toute satisfaction.

- Sous Linux et arborescence système.

Il est clair que dans ce cas la question des droits d'écritures accordés sur les répertoires et fichiers est un souci potentiel. Les JDK 1.4 bêta,

premières JDK à implémenter l'API *Preferences*, sauvegardaient sous `/etc/.java`, solution qui a été abandonnée par la suite. Ma JDK 1.6.0_02 sauvegarde dans son répertoire d'installation. J'ai installé cette JDK dans mon répertoire d'utilisateur, en `/home/fred/jdk-1.6.0`. Le répertoire de sauvegarde est finalement :

```
/home/fred/jdk-1.6.0/jre/.systemPrefs
```

Concrètement, comme j'ai les droits d'écriture dans le répertoire d'installation, une application se comportera à l'identique sous utilisateur `fred` et `root` (au sens Unix) si `fred` écrit le premier ses préférences système. Par contre si `root` (toujours au sens Unix) écrit le premier ses préférences système, les répertoires et fichiers lui appartiendront et un utilisateur commun tel que `fred` ne pourra pas ensuite modifier les préférences, ce qui est somme toute conforme à la logique. Dans le cas plus général où la JDK est installée pour tout le système, seule une application lancée sous le compte `root` aura le droit d'écrire des préférences système, ce qui est également conforme à la logique.

Pour conclure avec Linux, le comportement est grosso modo conforme à ce que l'on en attend à une exception près: si plusieurs JDK sont installées, il y aura autant de préférences système que de JVM employées pour lancer l'application. Alors que théoriquement les préférences système sont affiliées à un système hôte et non à une JDK.

- Sous Windows XP et arborescence utilisateur :

Les préférences sont stockées dans la base de registre, sous la clé

```
HKEY_CURRENT_USER\Software\JavaSoft\Prefs
```

pour la racine de l'arborescence. Si on change d'utilisateur Windows, le stockage a lieu sous la même clé, mais le système veille à ce que chaque valeur soit propre à l'utilisateur. Donc tout fonctionne bien, que l'utilisateur ait ou non les droits de l'administrateur.

- Sous Windows XP et arborescence système :

Les préférences sont stockées dans la base de registre sous la clé

```
HKEY_LOCAL_MACHINE\SOFTWARE\JavaSoft\Prefs
```

pour la racine de l'arborescence. Les clés sont communes à tout utilisateur Windows. On a donc bien des clés valables pour tout le système, tout va bien. Les préférences sont également communes aux différentes JDK installées, le contrat est parfaitement respecté. Par contre et c'est logique, une application Java qui veut écrire une préférence système doit être lancée depuis un compte ayant les droits de l'administrateur.

- Sous Windows Vista et arborescence utilisateur :

Les préférences sont sous la clé

```
HKEY_CURRENT_USER\Software\JavaSoft\Prefs
```

pour la racine de l'arborescence. Si on change d'utilisateur Windows, les préférences sont stockées sous la même clé, mais chaque valeur est propre à l'utilisateur. Donc tout fonctionne bien, comme sous Windows XP.

- Sous Windows Vista et arborescence système :

Nous rencontrons cette fois le mécanisme de virtualisation de la base de registre. A priori la JVM va essayer d'écrire les mêmes clés que sous Windows XP, mais Windows Vista redirige la sortie dans la base de



registre sous la clé

HKEY_CURRENT_USER\Software\Classes\VirtualStore\MACHINE\SOFTWARE\JavaSoft\Prefs

Comme on peut le voir illustré ci-dessus. La clé HKEY_LOCAL_MACHINE\SOFTWARE\JavaSoft existe toujours mais n'est pas utilisée. Nous avons là quelque chose de très particulier. Les valeurs des clés ne sont pas communes à tout utilisateur Windows, mais sont propres à chacun, en raison de ce mécanisme de virtualisation. Les préférences n'ont donc de système que le nom. En outre, le mécanisme de virtualisation permet à tout utilisateur, même sans droits administrateur d'écrire des préférences systèmes dans le registre, ce qui est contre intuitif. L'API Java Preferences est donc clairement cassée sous Vista. Il est vrai qu'à l'impossible nul n'est tenu. Il est fort probable que Sun proposera une parade dans une JDK à venir.

De tout ce qui précède, nous pouvons déduire que, quel que soit le système hôte, l'API *Preferences* est sujette à caution en ce qui concerne les préférences système, mais se comporte parfaitement et se révèle très pratique en ce qui concerne les préférences utilisateur, ce qui finalement est le principal.

5 Exporter ou importer des préférences

Il est possible d'exporter des préférences dans un fichier XML. Aucune difficulté ici. Voici un extrait de code qui exporte toute l'arborescence utilisateur :

```
Preferences pref = Preferences.userRoot();
File fu = new File("user prefs");
FileOutputStream fos = new FileOutputStream(fu);
pref.exportNode(fos);
```

Pour l'importation on procédera symétriquement, avec la méthode `importPreferences`.

6 Etre notifié d'une modification

La JVM peut notifier une modification de valeurs de clés ou même de création de nœuds, au moyen des événements `PreferenceChangeEvent` ou `NodeChangeEvent`. Comme pour d'autres événements

Java, le Design Pattern Observateur (cf. Programmez! 93) règne en maître. On s'enregistre comme écouteur et la JVM nous appelle en retour. Une seule réserve, le mécanisme ne fonctionne qu'au sein d'une seule JVM à la fois. Voici un exemple qui reçoit les notifications des modifications d'une valeur définie par notre premier exemple.

```
package programmez;

import java.util.prefs.*;

public class MonAppList implements PreferenceChangeListener {

    private static Preferences prefs;
    Boolean test;

    public MonAppList() {
        //prefs = Preferences.userNodeForPackage(
        //    programmez.MonApp.class);
        // ou bien
        prefs = Preferences.userRoot();
        prefs = prefs.node("/programmez");
        test = prefs.getBoolean("reseau", false);
        System.out.println("Reseau: " + test);
        prefs.addPreferenceChangeListener(this);
    }

    public void change() {
        if(test)
            prefs.putBoolean("reseau", false);
        else
            prefs.putBoolean("reseau", true);
    }

    public void preferenceChange(PreferenceChangeEvent evt) {
        String cle = evt.getKey();
        String valeur = evt.getNewValue();
        String noeud = evt.getNode().toString();
        System.out.println("Sur le noeud: " + noeud);
        System.out.println("Modification de la cle: " + cle);
        System.out.println("Nouvelle valeur: " + valeur);
    }

    public static void main(String[] args) {
        MonAppList mal = new MonAppList();
        mal.change();
    }
}
```

■ Frédéric Mazué
fmazue@programmez.com

Tous les jours : l'actu et le téléchargement
www.programmez.com

Introduction à la programmation avec Haskell

2^e partie

La programmation fonctionnelle est un atout maître dans la panoplie du développeur. Après avoir présenté Haskell le mois dernier, nous faisons plus amplement connaissance avec lui aujourd'hui, de manière ludique.

La programmation fonctionnelle est trop souvent reléguée au rang de curiosité universitaire, sans utilité pratique. Pourtant ce paradigme de programmation a tout ce qu'il faut pour séduire lorsqu'on est familiarisé avec lui. Plutôt que d'énumérer, de manière académique, les fonctionnalités du langage, cet article vous propose de voir, de manière ludique, comment résoudre de petits problèmes précis avec Haskell. Votre serveur espère ainsi montrer et surtout faire apprécier, par cette démarche, la supériorité de ce langage, en termes d'expressivité, de concision et d'élégance, sur les langages impératifs. Vous trouverez tous les exemples de code sur le Cd-Rom accompagnant le magazine, ou sur notre site.

1 Les outils

Un programme Haskell peut être interprété ou compilé en code natif. De l'humble avis de votre serveur, le meilleur outil est GHC, pour Glasgow Haskell Compiler. L'outil fournit aussi bien un compilateur, `ghc` lui-même, qu'un interpréteur du nom de `ghci`. Le code s'écrit avec votre éditeur préféré. Les fidèles lecteurs de *Programmez!* se douteront que je recommande avant tout le mode Haskell d'Emacs. Toutefois, il existe plusieurs plug-in pour Eclipse et un add-on Visual Haskell pour coder sous Visual Studio 2003 ou 2005.

2 Pour se faire la main

Sacrifions à la tradition et écrivons un "Hello World!":

```
main = do
  putStr "Hello World!\n"
  putStr "Programmez!\n"
  putStr "Abonnez vous\n"
```

Pour obtenir un exécutable avec `ghc`, on donne simplement la commande :

```
ghc -o hello HelloWorld.hs
```

Si notre programme était composé de nombreux modules, c'est-à-dire de nombreux fichiers sources, nous pourrions demander à `ghc` de les rechercher et compiler tous ainsi :

```
ghc -make -o hello HelloWorld.hs
```

En dépit de cette apparente simplicité, GHC est un environnement complet et complexe, dont la découverte dépasse le cadre de cet article. Pour interpréter notre code nous pouvons lancer `ghci` ainsi, en supposant que nous sommes positionnés dans le répertoire contenant le source :

```
ghci HelloWorld.hs
```

Le fichier `HelloWorld.hs` est alors chargé et analysé entièrement. Nous pouvons lancer le programme en demandant l'évaluation de la fonction `main` à l'invite de `ghci` :

```
main
Hello World!
Programmez!
Abonnez vous
```

On peut aussi commencer par lancer l'interpréteur `ghci` sans argument sur la ligne de commande, puis charger (ou recharger après modification) un source comme ceci :

```
:load HelloWorld.hs
```

Tous ces processus étant plus ou moins automatisés selon l'éditeur de votre choix. A ce stade, il est très important de noter que la syntaxe d'Haskell est bi-dimensionnelle, ce qui signifie que les indentations sont significatives et signalent des portées, comme avec Python par exemple. C'est pourquoi les instructions `"putStr"` de notre Hello World sont toutes tabulées de la même façon. La manière d'indenter est intuitive et sera illustrée par les exemples de cet article. Si l'on préfère, il est toutefois possible de délimiter les blocs de code par des accolades et de découper en séquence par des points virgules, comme en C par exemple. Voici à titre d'exemple un Hello World parfaitement affreux, mais parfaitement valide :

```
main = do
{
  putStr "Hello World\n" ;
  putStr "Programmez!\n"
  ; putStr "Abonnez vous\n"
}
```

Sous `ghci`, il est encore possible de demander directement l'évaluation d'expressions. Ainsi nous pouvons taper :

```
[1, 2] ++ [3, 4]
```

Ce qui sera évalué à :

```
[1,2,3,4]
```

Car nous avons demandé la concaténation des deux listes d'entiers en une seule. En programmation fonctionnelle, donc en programmation Haskell, la liste est une structure de données omniprésente.

3 Un ROT13 écrit en Haskell

Le ROT13 est un procédé rudimentaire de cryptage de texte. Comme le nom le suggère, il s'agit de décaler de 13 lettres la lettre à crypter. Ainsi A devient N, B devient O, N devient A et ainsi de suite. On ne code que les lettres de l'alphabet sans s'occuper des chiffres, des caractères accentués, de la ponctuation et autres symboles. La particularité de ce système réside dans le fait que 13 est la moitié de 26, qui est le nombre de lettres de notre alphabet. Ainsi, crypter en ROT13 un texte déjà crypté en ROT13 revient à rétablir le texte original. Il découle encore de cela que l'alphabet peut être vu comme circulaire, ou infini, A succédant à Z et ainsi de suite autant que l'on voudra.

Ecrire un programme de ROT13 est un jeu d'enfant qui met en relief quelques aspects du langage Haskell. Ainsi nous verrons les lettres de l'alphabet comme une liste de lettres ordonnées. En Haskell une lettre aura le type Char, or il se trouve que le type Char est un type énuméré. Ceci permet de construire des listes sans déclarer chaque élément :) Voici donc notre alphabet:

```
alphabet = ['A'..'Z']
```

Ce que nous n'avons pas encore pour notre alphabet est la circularité, c'est-à-dire une apparence infinie. Pour encoder la lettre Y par exemple nous devons pouvoir aller prendre la 13e lettre à partir de Y, ce qui n'est pas possible pour l'instant. Si notre alphabet était constitué de deux listes juxtaposées cela marcherait. Nous pourrions donc écrire :

```
alphabet = ['A'..'Z'] ++ ['A'..'Z']
```

Cela fonctionnerait, mais un programmeur Haskell averti fera mieux. Pourquoi, après tout, ne pas faire en sorte que le programme fonctionne aussi si le décalage n'est pas 13 mais 39 ? Allons nous concaténer manuellement 3 alphabets ? Non. Au lieu de cela, nous allons profiter de la puissance d'expression du langage qui permet de construire des ensembles de données infinis. Le mois dernier, dans notre présentation de la programmation fonctionnelle, nous avons déclaré l'ensemble des entiers naturels sous la forme d'une liste infinie.

```
ensembleN = [1..]
```

C'était puissant. Et nous avons vu que cela était possible grâce au mécanisme d'évaluation paresseuse du langage. Ce mécanisme fait uniquement ce qui est nécessaire, et seulement au moment où c'est nécessaire et évalué. Haskell permet aussi de construire des ensembles sur la base de fonctions récursives, c'est-à-dire des fonctions qui s'évaluent à partir d'elles-mêmes. Nous déclarons donc notre alphabet ainsi :

```
alphabet = ['A'..'Z'] ++ alphabet
```

Nous avons une fonction récursive non terminante qui s'évalue sous la forme d'une liste qui se concatène à elle-même autant que de besoin, mais pas plus, toujours en vertu du mécanisme d'évaluation paresseuse. Avec notre alphabet, notre programme sera un ROT13 ou un ROT39 ou un ROT65 tout aussi bien :) Pour écrire un programme complet nous devons encore accéder à un élément de la liste (notre alphabet). Cela se fait en Haskell au moyen de l'opérateur (!!). Voici le code qui parle maintenant de lui-même:

```
import Char

alphabet = ['A'..'Z'] ++ alphabet

rot13 c =
  let uc = toUpper c
      ordA = ord 'A'
  in alphabet !! (ord uc - ordA + 13)

main = do putStrLn $ map rot13 "Programmez"
```

Après avoir déclaré notre alphabet, nous déclarons une fonction rot13 qui code un seul caractère à la fois. Toutefois, nous voulons coder des chaînes de caractères. En Haskell une chaîne est une liste de caractères. Il suffit donc d'appliquer notre fonction rot13 à chaque élément de la liste. En Haskell comme avec tous les langages fonctionnels, il existe une fonction dite d'ordre supérieur qui prend en argument une fonction et une liste, applique la fonction à chacun des éléments de la liste, et retourne la liste des résultats. Cette fonction s'appelle *map*. Il est maintenant aisé de comprendre la fonction *main* de notre programme. Dans cette ligne, le signe \$ signifie que la fonction *putStrLn*, c'est à dire la sortie sur console d'une ligne, sera appliquée au résultat de *map rot13 "Programmez"*

Il n'aura pas échappé au lecteur attentif que le nom de son magazine préféré s'écrit avec un point d'exclamation. En l'état, notre fonction rot13 ne marche pas si elle est appliquée à autre chose qu'une lettre. Nous la ré-écrivons donc en utilisant une syntaxe particulière d'Haskell (syntaxe courante avec les langages fonctionnels): les gardes. Voici notre nouveau programme :

```
import Data.Char

alphabet = ['A'..'Z'] ++ alphabet

rot13 c
  | (c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z') =
    let uc = toUpper c
        ordA = ord 'A'
    in alphabet !! (ord uc - ordA + 13)
  | otherwise = c

main = do
  let s = map rot13 "Programmez!"
  putStrLn s
  putStrLn $ map rot13 s
```

Nous avons également modifié notre fonction main. Cette fois nous codons et décodons le texte "Programmez!" dans la même foulée. Pour cela le résultat intermédiaire est lié à une expression locale baptisée s.

4 Jouons aux cartes

Haskell dispose d'un système de type fantastique, que nous allons découvrir en écrivant quelques lignes de codes qui pourraient constituer les bases d'un jeu de cartes. Supposons que nous voulions modéliser un jeu de 32 cartes. Nous pouvons commencer ainsi :

```
data Carte = Sept | Huit | Neuf | Dix
           | Valet | Dame | Roi | As
```

Nous définissons ainsi un type de données utilisateur, `Carte`, et autant de constructeurs (`Sept`, `Huit`, etc.) pour ce type que de besoin. Au passage, on remarquera que les noms de types et de constructeurs commencent obligatoirement par une majuscule en Haskell. Un fichier source contenant cette déclaration pourrait être chargé dans `ghci`. Toutefois, si nous tapons par exemple `Roi` à l'invite de `ghci`, nous obtenons un message d'erreur :

```
No instance for (Show Carte)
```

Concrètement cela signifie que le système ne sait pas représenter le type sous la forme d'une chaîne de caractères pour affichage sur la console.

5 Ces types ont de la classe

En Haskell les types obéissent à des lois définies dans des classes de types. Une classe est un contrat. En tant que telles, les classes Haskell présentent une forte analogie avec les interfaces de Java ou C#. Mais à la différence de C# ou Java, les classes Haskell supportent l'héritage multiple. Pour qu'un type puisse être représenté sous la forme d'une chaîne de caractères, il doit être une instance de la classe `Show`. Nous verrons plus loin comment créer une instance de classe. Pour l'instant, nous nous contentons de ce que sait faire le système par défaut en ré-écrivant notre déclaration :

```
data Carte = Sept | Huit | Neuf | Dix
           | Valet | Dame | Roi | As deriving Show
```

Ne nous laissons pas tromper par le mot 'deriving'. `Carte` est maintenant une instance de `Show`, et non une classe dérivée. A ce stade, si nous tapons `Roi` à l'invite de `ghci`, nous constatons que le message d'erreur a disparu. Nous pouvons maintenant écrire un petit programme qui teste si une carte est "habillée" ou non :

```
data Carte = Sept | Huit | Neuf | Dix
           | Valet | Dame | Roi | As deriving Show

estHabille c =
  case c of Valet -> True
           Dame -> True
           Roi -> True
           _ -> False

main = do
  putStr $ show Dix ++ " "
  putStrLn $ show $ estHabille Dix
  putStr $ show Dame ++ " "
  putStrLn $ show $ estHabille Dame
```

Ce code n'a comme intérêt que de montrer le fonctionnement d'une instruction `case`.

On remarquera le caractère générique (`_`), en fin de test.

6 Des classes pour tout faire

Les instructions `case` manquent en général d'élégance. Si dans le code précédent, on pouvait tester l'égalité de `c` avec `Valet`, `Dame`, etc., on pourrait mettre des gardes à la place de l'instruction `case`. Pour qu'un test d'égalité puisse être effectué, le type doit être une instance de la classe `Eq`. Comme précédemment, profitons de ce que le système offre par défaut, et écrivons :

```
data Carte = Sept | Huit | Neuf | Dix
           | Valet | Dame | Roi | As deriving (Show, Eq)

estHabille c
  | c == Valet = True
  | c == Dame = True
  | c == Roi = True
  | otherwise = False

main = do
  putStr $ show Dix ++ " "
  putStrLn $ show $ estHabille Dix
  putStr $ show Dame ++ " "
  putStrLn $ show $ estHabille Dame
```

Notre type est maintenant instance de deux classes et notre code est plus propre. Mais il reste bien mauvais. Dans l'exemple `rot13`, nous avons construit une liste en profitant du fait que `Char` est un type énuméré. Nous pouvons faire de même avec Haskell :

```
data Carte = Sept | Huit | Neuf | Dix
           | Valet | Dame | Roi | As deriving (Show, Eq, Enum)

estHabille c = elem c [Valet .. Roi]

main = do
  putStr $ show Dix ++ " "
  putStrLn $ show $ estHabille Dix
  putStr $ show Dame ++ " "
  putStrLn $ show $ estHabille Dame
```

Nous commençons à avoir du beau code "haskellien". La fonction `estHabille` se réduit à une ligne qui teste l'appartenance de la carte à la liste des cartes de `Valet` à `Roi` :

7 Travailler avec des types complexes

On peut reprocher à ces exemples d'être loin de la réalité. Une carte ce n'est pas qu'une valeur, c'est aussi une couleur. Deux informations que l'on mettrait aisément dans une classe `Carte` avec un langage orienté objet. En Haskell, nous définissons des types composés dont nous nous servons directement pour construire et afficher tout le jeu de cartes sans même écrire une boucle. Voici comment :

```
data Couleur = Coeur | Pique | Carreau | Trefle deriving (Show, Enum)

data Figure = Sept | Huit | Neuf | Dix | Valet
            | Dame | Roi | As deriving (Show, Enum)
```



```
data Carte = Carte Figure Couleur deriving Show

jeu = [(Carte f c) | f <- [Sept .. As], c <- [Coeur .. Trefle]]

main = do putStrLn $ show jeu
```

Carte est maintenant un type composé. Lui-même et ses composants dérivant de Show, l'affichage est automatique. Le jeu de cartes est construit au moyen de ce qui s'appelle une liste compréhensive selon cet ordre: Sept Coeur, Sept Pique, etc. Si l'on considère que la déclaration de Figure peut s'écrire sur une seule ligne (la disposition adoptée ici l'est seulement pour le confort de lecture), la déclaration des cartes, la construction du jeu et son affichage tiennent en 5 lignes. Quel langage impératif est capable d'en faire autant ? :)

8 Implémenter une classe de type

On peut reprocher à l'exemple précédent de ne pas afficher les cartes sous une forme naturelle. À l'affichage "Carte Sept Coeur", on préférerait sans doute "Sept de Coeur". Concrètement, nous ne voulons plus nous contenter de l'implémentation de Show par défaut, mais au contraire en fournir une nous-mêmes. Nous devons d'abord supprimer 'deriving Show' de la déclaration de Carte. Et nous devons ensuite satisfaire au contrat défini par la classe Show. Voici la déclaration (ici simplifiée) de cette classe :

```
class Show a where
    show :: a -> String
```

Nous savons que Haskell est un langage qui supporte la généricité. Ici la lettre **a** veut dire un type quelconque. Nous lisons donc cette déclaration ainsi: tout type (quelconque) **a** est une instance de la classe Show s'il existe une fonction show qui reçoit ce type en argument et retourne une chaîne de caractères. C'est donc un jeu d'enfant d'implémenter l'affichage d'une carte. Le travail est réduit au minimum si nous profitons de ce qu'offre le système pour les types Figure et Couleur. Voici le nouveau code :

```
data Couleur = Coeur | Pique | Carreau | Trefle deriving (Show, Enum)

data Figure = Sept | Huit | Neuf | Dix | Valet
            | Dame | Roi | As deriving (Show, Enum)

data Carte = Carte Figure Couleur

instance Show Carte where
    show (Carte f c) = show f ++ " de " ++ show c

jeu = [(Carte f c) | f <- [Sept .. As], c <- [Coeur .. Trefle]]

main = do putStrLn $ show jeu
```

On remarquera la syntaxe particulière: notre fonction show est définie dans ce qui est déclaré comme étant une instance de Show. Moyennant quoi nous avons un affichage conforme à nos désirs :

```
[Sept de Coeur,Sept de Pique,Sept de Carreau, - etc.]
```

Apportons une dernière amélioration à nos cartes à jouer avant de nous quitter. Supposons que nous voulions tester l'égalité de deux cartes. Si nous écrivons :

```
data Carte = Carte Figure Couleur deriving Eq
```

le comportement par défaut fera qu'un Sept de Coeur par exemple sera différent d'un Sept de Pique. Mais dans la logique de notre jeu nous voudrions que tous les Sept soient égaux, tous les Huit soient égaux et ainsi de suite. Nous devons donc écrire nous même notre instance de la classe Eq qui est déclarée ainsi :

```
class Eq a where
    (==), (/=) :: a -> a -> Bool
    x /= y = not (x == y)
    x == y = not (x /= y)
```

Déclaration étourdissante s'il en est car chaque opérateur d'égalité ou d'inégalité renvoie à la déclaration de son homologue :) La raison est la suivante: dès que l'on a défini un opérateur, l'autre l'est ipso facto, comme on le constate dans l'exemple ci-dessous :

```
data Couleur = Coeur | Pique | Carreau | Trefle deriving (Show, Enum)

data Figure = Sept | Huit | Neuf | Dix | Valet
            | Dame | Roi | As deriving (Show, Enum, Eq)

data Carte = Carte Figure Couleur

instance Show Carte where
    show (Carte f c) = show f ++ " de " ++ show c

instance Eq Carte where
    (Carte f c) == (Carte f' c') = f == f'

jeu = [(Carte f c) | f <- [Sept .. As], c <- [Coeur .. Trefle]]

main = do
    let carte = Carte Huit Pique
    putStrLn $ show $ carte == (jeu !! 0)
    putStrLn $ show $ carte == (jeu !! 1)
    putStrLn $ show $ carte == (jeu !! 4)
    putStrLn $ show $ carte /= (jeu !! 4)
    putStrLn $ show $ carte /= (jeu !! 5)
    putStrLn $ show $ carte == (jeu !! 8)
    putStrLn $ show $ carte /= (jeu !! 8)
```

L'implémentation de == se contente de tester l'égalité des composants Figure d'une carte, en se basant sur le comportement par défaut. On peut difficilement faire plus simple. Si l'on prend un peu de recul, on se rend compte que l'on a réalisé un énorme travail de modélisation de cartes à jouer en seulement huit lignes de code (hors la fonction main qui ne contient que des tests). Sans rencontrer la moindre difficulté, en exprimant les choses de manière très naturelle. Il est vraiment très fort ce Haskell !)

■ Frédéric Mazué - fmazue@programmez.com

ACTUS

Vers la maîtrise graphique



Le logiciel de traitement d'image PTrack 4.1 a désormais sa "learning edition", version d'apprentissage téléchargeable sur le site de l'éditeur. Seule l'exportation des données images et vidéos (camera) est limitée, l'ensemble des outils est complet. Si vous ne le connaissez pas, Ptrack est un soft de travail 3D étonnant, puisque la plupart de ses effets de modélisation peuvent se faire à la main, disons à la souris, avec des brosses dédiées, dont certaines détectent d'elles-mêmes les contours ou surfaces à traiter. La gestion des animations est aussi très poussée, avec mixage de sources très diverses. www.pixelfarm.com

Crédit d'impôt pour les jeux vidéo : enfin !

Vous pensiez que c'était fait ? Eh bien non. La Commission Européenne, après presque un an d'enquête, vient seulement de donner son aval à la proposition française de crédit d'impôt pour la production de jeux vidéo. La mesure devrait donc pouvoir enfin booster les budgets de création des sociétés hexagonales, si toutefois cette ouverture financière lui est vraiment consacrée. La loi aménagée a été votée le mois dernier, elle devrait être effective courant février ou mars. Vous pouvez trouver plus de précisions sur le site de l'Association des producteurs d'œuvres multimédia, APOM.org, dont l'action a été décisive.

Le Game Design pris au sérieux

Game Designer, c'est maintenant un métier ! Le diplôme de Game Designer NII de l'école Isart Digilat est désormais homologué, en

Bac + 4. La Prépa Game Design correspondante commence courant février, elle est accessible avec le Bac. L'équipe pédagogique est composée d'acteurs actuels de l'industrie du jeu vidéo. La formation peut se compléter d'un cursus Web, en deux ans, et de cours du soir consacrés à l'animation 2D et 3D. Bref, tout pour faire de vous la prochaine recrue d'élite d'Ubi Soft ou d'Electronic Arts. Ou alors Blizzard ? J'hésite... Plus de renseignements sur www.isartdigital.com

Neuro-ludisme

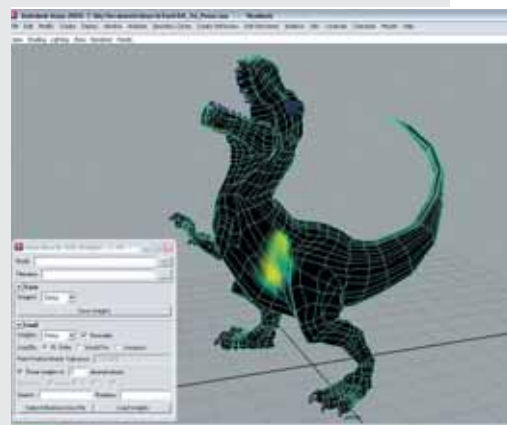


Après vous avoir conseillé les "serious games" (jeux éducatifs pour administrations et organismes) comme voie de développement et de fortune, nous vous recommandons aujourd'hui de vous lancer dans les "neurogames". Le secteur des neurosciences, consacré à l'étude et au développement des capacités du cerveau, attire en



effet déjà tous les capital-risqueurs américains, avec même un secteur du Nasdaq aujourd'hui dédié. Quant à Nintendo, il a lancé la mode en 2007 en vendant plus de 4 millions d'exemplaires de son fameux jeu sur DS d'Entraînement cérébral du Docteur Kawashima (neuroscientiste japonais médiatique et assez contesté). Les dérivés se multiplient donc, et accumulent les excellents chiffres de ventes... ou de connexions. Car l'intelligence s'éduque et se joue désormais aussi en ligne ! Allez chercher des pistes et des idées du côté de ces sociétés nouvellement créées comme Happy Neuron (sic), www.happyneuron.fr, Positscience (www.positscience.com), le très mystique Lumosity (www.lumosity.com) ou encore l'europpéen et plutôt sérieux Mindfit (www.mindfit.be). Développement et investissement minimaux pour profit maximal, c'est à coup sûr l'intelligence !

Effets de muscle chez Maya



Autodesk vient d'intégrer le module "Muscle" à la version 2008 de Maya. Le programme, créé à l'origine par Comet Digital, permet de traiter les animations musculaires des personnages ainsi que les effets de peau. On trouve bien sûr tous les outils de tracé initial nécessaires, avec programmation d'effets de déformation selon une galerie de mouvements et d'attitudes pré-programmées, issues du logiciel ou des utilisateurs (forum). Bref, Maya se soucie aussi désormais de gérer la puissance des héros baraqués... ou des dinosaures géants !

Un portable pour joueurs et développeurs pros

MSI, plus connu en tant que fabricant de cartes-mères et de cartes graphiques, s'est implanté sur le marché des PC portables depuis novembre dernier avec une gamme à part, essentiellement dédiée aux joueurs. Le dernier né, G600, baptisé à raison " Extrême ", est en effet une perle rare. D'abord, il est équipé comme une configuration ludique desktop haut de gamme : processeur(s) Intel Core Duo T7500, 2 Go de Ram en 667 mHz, disque dur de 250 MO, et surtout une Geforce 8600 GTM pour le graphisme, au service d'un écran en 1680 x 1050 de résolution, soit quasiment de la full HD ! D'ailleurs, la connectique offre la sortie HDMI nécessaire à ce niveau de performance pour un affichage hallucinant sur téléviseur HD. Ensuite, l'Extrême est conçu pour l'overclocking : une simple touche à côté du clavier permet de doser ce TDE, ou Turbo Drive Engine. Les processeurs peuvent ainsi monter à 2,66 Ghz. C'est la plus grande innovation, car on sait que les problèmes de refroidissement sur portable sont spécialement pointus. Le désagré-



ment à payer ici est un ventilateur qui se fait franchement entendre en phases de jeu poussées, et aussi, avec l'ensemble des équipements, un transfo d'alimentation imposant, dont les dimensions avoisinent les 2/3 du portable lui-même ! L'autonomie reste cependant supérieure à 2 heures : pas mal. Avec un prix qui reste finalement raisonnable (environ 1500 €), MSI ouvre donc une nouvelle voie hyper-ludique aux ordinateurs portables. Une stratégie exactement antinomique à celle d'Asus, et de son fameux Eee (toujours attendu en France), ultra-mini-portable à performances bureautiques et prix hyper-réduit (250 €) environ. En tout cas, avec ces deux outils, on peut voir là la mallette informatique idéale du nouveau développeur : un mini-PC pour le code, une config de pro pour le graphisme, l'exécution, et les démos. Le développement ludique nomade est né.

Nintendo joue la carte du risque

Nintendo a choisi : les cartes MicroSD officielles pour la Nintendo DS seront finalement commercialisées, et elles devraient être disponibles en Europe au moment où vous lisez ces lignes. Ces cartes spécifiques, de 512 Mo, se glissent dans un boîtier en forme de cartouche, qui s'insère alors très banalement dans la console. Depuis un PC, l'utilisateur peut y copier et y manipuler aisément tous types de fichiers.



Le choix était cornélien pour l'éditeur nippon : d'un côté, oublier cette carte, c'était priver la DS d'étendre ses capacités à celles d'un authentique baladeur multimédia (photos, musiques, films), et surtout renoncer au futur système de téléchargement de jeux et d'applications Nintendo, en place d'ici peu. De l'autre côté, lancer la carte, c'était booster encore le potentiel de la DS, mais en même temps ouvrir grand la porte au piratage, déjà fort actif sur la console portable. En effet, les cartes Nintendo ne sont jamais que la version officielle des fameux " linkers ", ces boîtiers à carte fabriqués en Chine qu'on trouve depuis plusieurs mois et qui, sans aucun flashage de la console, permettent d'y copier des jeux téléchargés sur Internet (ou, bien sûr, du contenu multimédia). Alors, autant guérir le mal par le mal ?

En tout cas, Nintendo vient de faire un choix totalement inhabituel pour lui, celui de " l'insécurité " : une réaction logique face au piratage déjà effectif, mais surtout la preuve directe que le marché des jeux en téléchargement vaut désormais tous les sacrifices ! La carte Nintendo officielle coûtera aux alentours de 30 €.

RPG Maker VX : faites du Zelda (et mieux)



Depuis 2005, la vedette incontestée du développement indépendant de RPG à la japonaise était le logiciel RPG Maker XP, de la société Enterbrain. Une véritable merveille de simplicité pour créer des aventures à la Zelda ! En ce début 2008, un successeur vient enfin de sortir, RPG Maker VX, si frais que sa traduction en anglais et en français sont encore à peine disponibles (adaptations de fan). Quoi de neuf avec VX ? Rien dans un premier temps, et heureusement, puisque la structure du programme et son interface miraculeuse sont conservées. Les spécialistes apprécieront cependant le *tlng* automatique, qui fait que le programme " ajuste " de lui-même toute

surface de paysage aux tuiles déjà existantes. Les objets ont aussi maintenant leur menu contextuel direct, de telle sorte qu'affecter une condition d'ouverture ou de contenu à un coffre sera aussitôt fait avec une équation booléenne et le jeu de variables nécessaires. L'affichage a aussi été optimisé : le scaling entre les divers éléments est plus large et plus facile à configurer, l'animation monte à 60 fps si souhaité, et une fonction " glow " offre désormais une prévisualisation instantanée des effets de lumière. Enfin, côté code, on a restreint les possibilités immédiates en création contextuelle, et élargi les outils annexes – excellent choix par rapport à la version XP, trop lourde en fonctions d'accès permanent. Bref, une grande réussite, à découvrir en japonais (!) sur <http://tkool.jp> ou, plus raisonnablement, avec versions traduites et forum, sur www.relite.org ou <http://rpgcreative.net>. La version téléchargeable est assez limitée, mais éloquente sur les possibilités de VX. De toutes façons, l'achat n'est pas un investissement ruineux : 100 \$ environ !

Windows Server 2003 et 2008 tome 1



- **Difficulté :** ***
- **Editeur :** Micro Application
- **Auteur :** collectif
- **Prix :** 57,20 €

Vous êtes administrateur d'un serveur Windows ou simplement intéressé par en savoir plus sur le prochain Server 2008 ? Ce pavé de 1000 pages propose un tour du propriétaire en mettant l'accent sur Server 2003 : installation, déploiement, fonctionnement avec Vista, déploiement de Vista sur les postes, utilisation de PowerShell ou encore la maintenance. Server 2008 est décrit principalement pour ses nouveautés, ensuite les auteurs explorent ActiveDirectory et la sécurité. Ceux qui attendent une perspective complète sur Server 2008 seront déçus, ceux qui veulent installer et mieux utiliser Server 2003 seront par contre ravis. Attendons le 2e tome...

Génériques et collections Java



- **Difficulté :** ***
- **Editeur :** O'Reilly
- **Auteur :** collectif
- **Prix :** 35 €

Avec Java, on peut depuis Java 5 utiliser les notions de génériques et de collections. Deux concepts particulièrement pratiques et puissants pour le développeur. Mais encore faut-il les maîtriser. Les auteurs abordent les design patterns, l'interface collection, la classe collection ou encore les bases des types génériques. Illustré de très nombreux exemples de code, vous deviendrez imbattables sur ces concepts java !

Introduction au design web



- 3e édition
- **Difficulté :** **
- **Editeur :** O'Reilly
- **Auteur :** Jennifer Niederst Robbins
- **Prix :** 45 €

Que vous soyez débutant ou que vous souhaitiez mettre à jour vos connaissances, ce livre vous donnera le coup de pouce qui vous manquait pour créer des pages web modernes.

Au travers de nombreux exercices et exemples, cet ouvrage explique comment créer des liens, ajouter des images, se servir intelligemment des standards, des feuilles de style, etc. Il couvre tout ce qui concerne le traitement des images et la mise à disposition de vos pages sur le Web. Ce livre se veut comme un guide pour le développeur web et graphiste. On y parle CSS, HTML, XHTML. Si vous êtes déjà expert en la matière, passez votre chemin. Pour ceux qui veulent approfondir des éléments CSS ou se rappeler quelques fondamentaux, n'hésitez pas !

Ergonomie web



- **Difficulté :** **
- **Editeur :** Eyrolles
- **Auteur :** Amélie Boucher
- **Prix :** 32 €

Qu'est ce que l'ergonomie web ? Si vous vous posez la question, ce livre vous donnera les clés pour comprendre et mettre en œuvre. L'auteur débute par la bonne et mauvaise ergonomie avant de comprendre de quelle manière on la travaille. On remet en perspective l'utilisateur, qui reste le focus du créateur web. Puis, on nous assène les 12 règles de l'ergonomie web. Il est dommage que le livre reste axé web sans aborder un contexte de plus en plus important : RIA, RDA, Air, Flex, Silverlight, XUL, XAML, etc. Car ces plates-formes et technologies impactent considérablement l'ergonomie et le design.

Débuter en Javascript

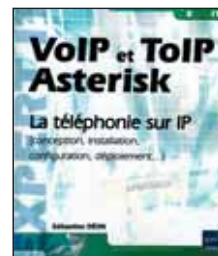


- **Difficulté :** ***
- **Editeur :** Eyrolles
- **Auteur :** Shelly Powers
- **Prix :** 29,90 €

Javascript demeure un langage obscur pour de nombreux développeurs. Difficile à maîtriser, Javascript est parfois utilisé dans Ajax, ce qui le rend indispensable. Pour ceux qui ne le connaissent pas encore, ce livre se propose de les initier à ce langage. Cet ouvrage présente en des termes simples toute la syntaxe de base du langage et propose une description détaillée des principales bibliothèques existantes. Les nombreux exemples concrets mettent en avant les meilleures pra-

tiques de programmation, en insistant sur tous les pièges à éviter, en particulier tous les problèmes liés à la compatibilité.

VoIP et ToIP Asterisk



- **Difficulté :** ***
- **Editeur :** Eni éditions
- **Auteur :** Sébastien Déon
- **Prix :** 39 €

La VoIP est un des thèmes majeurs des nouvelles formes de communication dite unifiée. La VoIP a l'ambition de réduire les coûts et de proposer une alternative aux infrastructures actuelles. On revient tout d'abord à l'écosystème de la VoIP : les termes, l'infrastructure, les architectures, les codecs et les notions de qualité. Le second chapitre aborde plus spécifiquement le problème de l'architecture, puis on passe au plat de résistance avec le déploiement d'Asterisk. À partir de là, l'auteur décrit, étape par étape, la création de son environnement VoIP. Très intéressant pour bien démarrer avec Asterisk. Surtout, on trouve les scripts de paramètres, de configurations.

Collection Microsoft Press

- **Difficulté :** ***
- **Editeur :** Dunod
- **Auteur :** divers

SharePoint 2007 : Pratique et illustré de nombreux exemples, ce guide contient des informations détaillées sur toutes les fonctionnalités du logiciel, des conseils d'experts, des procédures et des astuces pratiques pour exploiter SharePoint Server 2007 dans les meilleures conditions.

PowerShell : outil qui permet aux administrateurs système d'automatiser et d'accélérer leurs tâches d'administration. Il offre aussi des fonctionnalités simplifiant l'écriture de scripts. Ce livre vous propose une plongée totale dans cet environnement. Inclut de très nombreux scripts !

Écrire du code sécurisé pour Windows Vista : attention, voici une bible pour tout développeur Windows ! Il aborde les nouvelles API, les fonctions avancées, l'utilisation par exemple de l'UAC ou du bitlocker. De nombreux codes C# permettent de mettre en œuvre immédiatement tout cela !

SERVEUR ECO D'AMEN UNE TECHNOLOGIE EN HARMONIE AVEC L'ECOSYSTÈME.



A partir de
49 € HT/mois **
soit 58,60 € TTC/mois

**NE CHOISISSEZ PLUS ENTRE
PERFORMANCE ET ÉCOLOGIE.
SERVEURS ECO D'AMEN :
DE 1 À 3 Go DE RAM,
55 % D'ÉMISSION DE CO₂
EN MOINS PAR AN.**

<http://eco.amen.fr>

- AMD Athlon X2 3400+ ou BE-2350
- De 2x1,8GHz à 2x2,1GHz
- Interface Plesk 8.2 – 10 domaines
- OS : Fedora Core 7, Ubuntu 6, Debian 4, Windows Server 2003
- De 1 à 3 Go de RAM
- Disque dur de 80 Go à 160 Go
- Trafic mensuel : 1 To
- 1 à 2 adresses IP
- Inclus : Reboot, Restore et Recovery
- En option : Amen DataBackup 10 Go (Inclus sur ECO 3000)



NUMÉRO GRATUIT : 0800 740 935 ou www.amen.fr

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

Votre potentiel, notre passion.™
Microsoft

ROME NE S'EST PAS FAITE EN UN JOUR.
VOS DÉVELOPPEURS, EUX, ONT UN MOIS PILE.

RELEVEZ TOUS LES DÉFIS



Votre défi : bâtir de grands projets à 300 000 km/s.
Vos armes : communiquez et travaillez mieux ensemble
avec Visual Studio™ Team System.
Plus d'informations sur www.releveztouslesdefis.com

McANN

© 2007 Microsoft Corporation. Tous droits réservés. Microsoft, Visual Studio, le logo Visual Studio, et « Votre potentiel, notre passion.™ » sont des marques de Microsoft déposées et/ou utilisées aux États-Unis et/ou dans d'autres pays.

Microsoft
Visual Studio