



Windows® 8

de A à W 2e PARTIE

- Les développeurs jugent Windows 8 et Visual Studio 2012
- Optimiser le chargement des applications
- Migrer le code Windows Phone 7

QR Code le tag à tout faire !

Outils

Virtualisation

L'outil magique
du développeur

Smartphone

Android 4.1 :
l'accélération
graphique par défaut

Java

Développez
une solution NFC !

Linux

User et abuser
des scans de port



N°1 EN FRANCE
WINDEV®

18

JUSQU'AU 18 DÉCEMBRE

**OPÉRATION POUR
1 EURO DE PLUS**

Votre code est compatible :

- Windows 8 (7, XP, ...)
- Linux
- Mac
- Internet
- Intranet
- Windows Mobile & CE
- Windows Phone
- Android
- iPhone et iPad.

Faites-vous plaisir!

116 CM

ULTRA FIN
HD 1920 X 1080
3D

UTILISABLE COMME
MONITEUR PC

Utilisable comme
moniteur PC: imaginez
vos présentations
sur cet écran géant!

Offre réservée aux sociétés, administrations et professions libérales, en France métropolitaine.

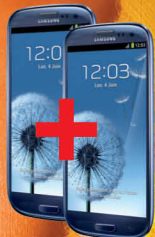
Logiciel et matériel peuvent être acquis séparément; chaque logiciel au tarif catalogue de 1.650 Euros TTC, le téléviseur au tarif de 1.050 Euros TTC, chaque Galaxy Tab au tarif de 599 Euros TTC, chaque Galaxy S3 au tarif de 670 Euros TTC (tarifs modifiables sans préavis). Voir tous les détails, tous les tarifs et des vidéos sur www.pcsoft.fr.

Aucun abonnement n'est à souscrire pour bénéficier de cette offre.

Elu «Langage le plus productif du marché»

Fournisseur Officiel de la Préparation Olympique

OU



OU



OU choisissez 2 Tablettes Samsung Galaxy Tab 2 10.1, OU 2 Smartphones Samsung Galaxy S3, OU encore 1 PC portable Dell, OU 1 PC tactile Dell OU 1 station de travail Dell

**COMMANDEZ
WINDEV®**

ou WINDEV Mobile 18 ou WEBDEV 18

**ET RECEVEZ
POUR 1 EURO DE PLUS**

**CE SUPERBE
TELEVISEUR SAMSUNG**

116 cm; LED haute définition 1920 x 1080, 200Hz, ultra fin, 3D, 2 paires de lunettes 3D, Internet, Wifi, Skype, 3 HDMI, 3 USB,...

Référence : UE46ES6300

WINDEV est LE logiciel qui permet de créer les logiciels.

Cette plateforme professionnelle permet de créer tous les types de logiciels, pour tous les domaines, pour tous les types de matériels, dans tous les environnements, pour toutes les volumétries.

Votre code, vos fenêtres, vos données, vos rapports,... sont compatibles.

Vous aussi, développez 10 fois plus vite pour toutes les plateformes.



www.pcsoft.fr



978
NOUVEAUTES

CUDA 5, parallèle mon amour !

Attendue avec impatience ou non, la 5e version majeure de CUDA est disponible depuis peu. Nvidia, son promoteur, ne veut pas laisser le développement parallèle ou GPGPU lui filer entre les processeurs graphiques.



Récemment nous parlions des réticences, plus ou moins visibles, de Nvidia envers OpenCL, technologie concurrente. Pour le fondeur, il s'agit de rester un acteur du parallélisme et surtout du GPGPU même si des technologies orientées systèmes, avec intégration matérielles telles que OpenCL ou C++ AMP font clairement de l'ombre à CUDA qui reste lié à ses processeurs graphiques.

CUDA 5 (encore en pré-version) doit apporter de nouvelles améliorations de performances parallèles, de nouvelles librairies de développement. Les principales nouveautés annoncées sont :

- parallélisme dynamique : l'objectif est de permettre dynamiquement à un thread GPU de créer de nouveaux threads pour optimiser les flux et les traitements. En minimisant les va-et-vient avec le processeur central, le parallélisme dynamique simplifie considérablement la programmation parallèle. De plus, cela permet une accélération via le GPU d'une gamme plus large d'algorithmes répandus, tels que ceux utilisés dans les applications d'« adaptive mesh refinement » et de simulation

numérique des fluides. Cette technique s'applique aux architectures Fermi et Kepler

- Librairies exécutables par la GPU : CUDA BLAS permet d'utiliser le parallélisme dynamique pour ses propres librairies exécutables par la GPU. Cette fonction doit permettre de faciliter l'usage de la GPU dans les applications.
- selon Nvidia : « La capacité de « création de lien avec un objet » offre un processus efficace et habituel pour le développement des applications sur GPU importantes, en permettant aux développeurs de réunir de nombreux fichiers sources CUDA dans des fichiers objets séparés et de les relier à de plus grandes applications et bibliothèques. »
- GPUDirect : cette technologie doit minimiser les goulots d'étranglement qu'engen-

drent la mémoire système. Le but est de garantir une bonne communication entre la GPU et les autres systèmes, sous-systèmes et les bus PCI-E. DirectGPU assure un accès direct à la mémoire entre les cartes réseaux et la GPU. Elle doit aussi réduire la latence des messages MPI Send / Recv

- Nsight pour Eclipse : autre nouveauté, l'environnement Nsight en environnement Eclipse. Cet IDE dédié doit faciliter, ou tout le moins aider, le développeur à coder, déboguer, mettre au point le code CUDA. L'outil fonctionne sur Linux et OS X. « De plus, la re-création automatique de code facilite le transport de boucles de processeur central vers les noyaux CUDA. Un système d'analyse professionnel intégré fournit une analyse des performances automatisée et une assistance pas à pas afin de réparer les goulots d'étranglements dans le code ; tandis que la mise en relief de la syntaxe facilite la différenciation entre le code du processeur graphique et celui du processeur central », rajoute l'éditeur.

CUDA 5 ne modifie pas le modèle de développement que l'on connaissait déjà

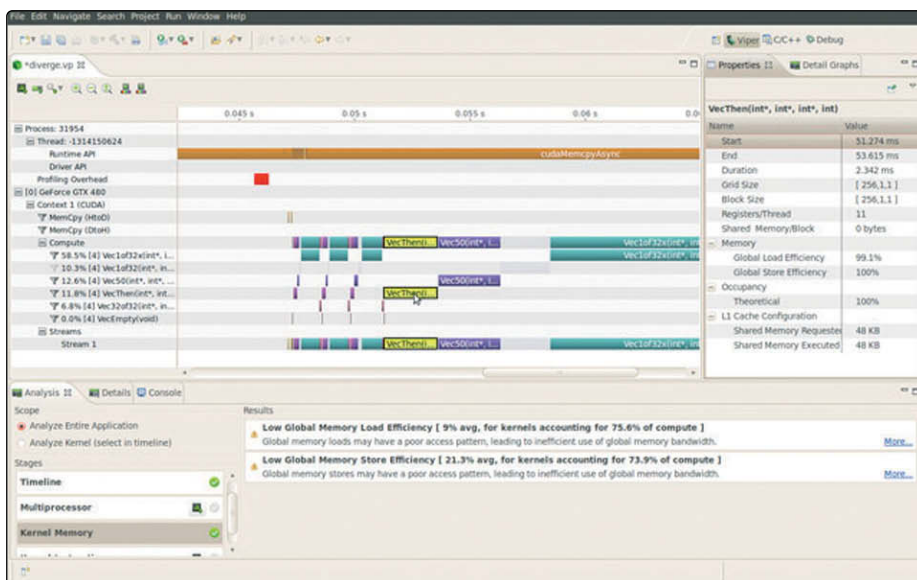
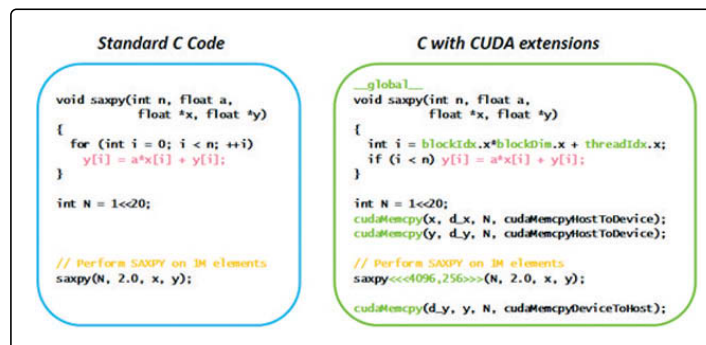
avec CUDA. Il faudra adapter le code entre les GPU Fermi et GPU Kepler. Le parallélisme dynamique peut offrir une nouvelle souplesse de mise en œuvre. On apprécie que la partie compilation ait été revue pour être moins rigide sur les sources des noyaux GPU à inclure, et tout avoir dans un fichier d'ensemble.

Nvidia en profite pour proposer un nouveau site de ressources CUDA pour faciliter la recherche et l'accès aux ressources techniques (outils, exemples, documentations) aux développeurs.

Site : <http://docs.nvidia.com>

Site CUDA :

<http://developer.nvidia.com/category/zone/cuda-zone>





BUZZWORD

CUDA 5, parallèle mon amour !3

ACTUS

En bref6



11

MATÉRIEL

Et si vous pensiez aussi à l'onduleur ?11

DÉVELOPPEUR DU MOIS

Christophe Peugeot :
Du CAP au métier
de développeur14

10

VEILLE TECHNO

Démystification
des QR codes24

La conquête des
applications mobiles
hybrides et l'émergence
d'un nouveau
standard
multi OS28



24

VIRTUALISATION

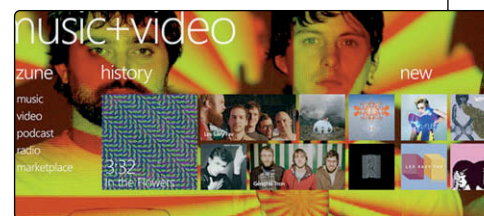
L'amie du développeur

Une offre large et pour tous les goûts31
Virtualisation sur OS X
pour développer sur Windows32



DOSSIER Windows® 8 de A à W

2^e partie



• Témoignages :

Olivier Conq : entre Visual Studio et Scrum37
Nathalie Belval39
Nathanael Marchand est Wahou !40

• Créez une application Windows 8 avec Visual Studio & TFS 2012 (partie 6) :

Les tests41

• Optimisation du lancement d'une application
Windows 8 en XAML/C#44
• Migrer une application Windows Phone 7 vers
Windows 848
• Portable Class Library
avec Visual Studio 201251



CARRIÈRE

Testeur : un métier réellement d'avenir.....54

Je DÉBUTE avec...

Comprendre le refactoring56

cas d'usage

Android 4.1 : le fonctionnement du Project Butter..59

PRATIQUE

Petit aperçu du développement SharePoint 2013 ..62

Utilisation de la technologie

NFC au sein d'une application Android66

Démystifier le développement

des plug-ins d'Eclipse70

Optimisation des performances

sur Windows Phone (2e partie).....75

webmaster

Scans de port : les comprendre,

les utiliser, s'en prémunir78

Chronique d'un bug dû à l'utilisation

des délégués en C#82



Editeur : Go-2 sarl, 21 rue de Fécamp 75012 Paris -
diff@programmez.com. - **Rédaction** : redaction@programmez.com - **Directeur de la Rédaction** : Jean Kaminsky. **Rédacteur en Chef** : François Tonic - ftonic@programmez.com. **Ont collaboré à ce numéro** : F. Mazué, S. Saurel. **Experts** : D. Vandevide, T. Leriche-Dessirier, S. Cordonnier, O. Conq, P. Didiergeorges, V. Fabing, J. Thiriet, B. Baldacci, J. de Oliveira, F. Bellahcene, M. Fansi, K. Alnjres, D. Hassoug, G. Bouveret, J. Alles, C. Gajdon, A. Weil, P. Marois, R. Bucin.

Illustration couverture : © istockPhoto/Akusan

Publicité : Régie publicitaire, K-Now sarl.

Pour la publicité uniquement : Tél. : 01 41 77 16 03 -
diff@programmez.com. Dépôt légal : à parution -
Commission paritaire : 0717K78366 ISSN : 1627-0908.
Imprimeur : S.A. Corelio Nevada Printing, 30 allée de la
recherche, 1070 Bruxelles Belgique.

Directeur de la publication : J-C Vaudecrane



Sauvons le NFC... ou pas !

La technologie NFC est le nec plus extra des smartphones, et donc, tout bon téléphone doit avoir sa puce NFC et sa fonction de paiement mobile et sans contact. A mort le iPhone 5 qui ose faire

l'impasse sur cette si belle technologie !

Mais à quoi sert une technologie si elle n'est pas réellement utilisée au quotidien, que les utilisateurs ne s'en servent pas et que son usage courant reste très limité ? Pas à grand-chose ! Et c'est le constat qu'a fait Osama Bedier de Google il y a quelques jours. Expert en paiement mobile, il prévoit que le NFC ne sera pas mature, dans l'usage, avant 3 à 5 ans ! Il y a bien des terminaux de paiements compatibles, mais en France ils sont peu nombreux et il faudra du temps avant que les commerçants s'en équipent. C'est aussi pour cela que Apple a sans doute estimé qu'il était trop tôt pour intégrer cette technologie, et met en avant une autre approche, Passbook, qui permet de stocker tickets, billets, cartes de fidélité, etc. Plusieurs grandes surfaces (Auchan, Leclerc) s'y sont lancées. Mais en France notamment, le problème vient encore une fois du manque de services compatibles.

La même remarque peut se faire sur le QR Code. Nous sommes plutôt enthousiastes sur cette technique, simple et rapide même si cela pose des problèmes de sécurité. Mais là encore, son usage a du mal à décoller en France par une généralisation encore limitée de ces codes, et surtout par le manque d'applications sachant les lire par défaut sur les téléphones et tablettes, voire sur nos PC. Mais l'usage doit faire ses preuves. Peut-être les développeurs vont pouvoir aider ?

Impossible de ne pas évoquer Windows 8, disponible depuis quelques jours. Pour Microsoft, ce nouveau système est un enjeu important et en particulier sur les changements radicaux de l'interface utilisateur et la notion de Modern UI. Mais cette interface, pensée pour les terminaux mobiles au départ, provoque une rupture assez radicale sur le desktop par rapport à Windows 7.

L'industrie du PC vit aujourd'hui une période difficile et Windows 8 n'est peut-être pas le sauveur annoncé, même si les ventes devraient être bonnes pour la fin de l'année. A condition que le marché redémarre avec vigueur, en quelques semaines. Le patron d'Intel, Paul Otellini, a du faire un rétropédalage après son « *Windows 8 n'est pas prêt* » qui a fait le tour du monde. Cependant, pour Microsoft, l'enjeu n'est pas tant Windows 8 que Windows Phone 8 et Windows RT avec les Nokia Lumia et les tablettes Surface. L'éditeur partant de très bas, il ne peut que faire mieux !

Mais Apple aura savonné la planche dès le 23 octobre avec l'iPad mini et Google avec Nexus, annoncé le 29 Octobre, le jour même du lancement de Windows 8 et Windows Phone 8 : les deux mastodontes de la mobilité veulent tout faire pour limiter le succès des nouveaux Windows.

La situation a bien changé depuis Windows 7 et surtout Windows XP. Et le moindre faux pas sera exploité, ou provoqué, par les concurrents.

Ce n'est pas RIM que dira le contraire, lui qui espère tant de son futur BlackBerry 10, ni même Nokia qui mise tout sur Windows Phone 8...

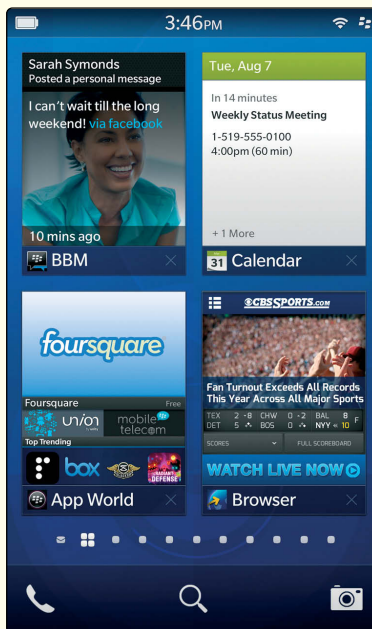
François Tonic - Rédacteur en chef
ftonic@programmez.com

Abonnement : Programmez, 17, Chemin des Boulangers, 78926 Yvelines Cedex 9 - Tél. : 01 55 56 70 55 - abonnements.programmez@groupe-gli.com - Fax : 01 55 56 70 91 - du lundi au jeudi de 9h30 à 12h30 et de 13h30 à 17h00, le vendredi de 9h00 à 12h00 et de 14h00 à 16h30. **Tarifs** abonnement (magazine seul) : 1 an - 11 numéros France métropolitaine : 49 € - Etudiant : 39 € - CEE et Suisse : 55,82 € - Algérie, Maroc, Tunisie : 59,89 € Canada : 68,36 € - Tom : 83,65 € - Dom : 66,82 € - Autres pays : nous consulter. **PDF :** 30 € (Monde Entier) souscription exclusivement sur www.programmez.com

» RIM se remotive

Tout d'abord, une nouvelle version système pour Playbook, la 2.1. Cette version augmente la sécurité des données et améliore la fonction Print to Go. BlackBerry PlayBook OS 2.1 offre de nouvelles fonctionnalités qui simplifient le déploiement des tablettes dans l'entreprise avec BlackBerry Mobile Fusion. Il s'agit notamment de la possibilité de se connecter à distance au serveur BlackBerry Mobile Fusion, du support pour l'authentification certifiée d'ActiveSync, et de nouvelles règles de sécurité informatiques. La mise à jour est disponible depuis la mi-octobre. Mais le plus intéressant concerne le futur BlackBerry 10 qui n'arrivera que début 2013. Les outils de développement ont été mis à jour (mais restent en pré-version). Cela concerne les développements C++, HTML 5, Qt, Adobe Air et Android Java. Le Core Application inclut de nouveaux services (calendrier, contacts, etc.). La partie native dispose d'un plug-in Visual Studio. Et on bénéficie aussi d'un simulateur Ripple pour PhoneGap.

Site : <https://developer.blackberry.com/>



plateforme comprend un moteur de remédiation (en attente de brevet) qui fournit aux développeurs des conseils précis spécifiques aux défauts pour les aider à résoudre les problèmes de sécurité correctement et efficacement, sans connaissance particulière.

» La librairie Mesa (implémentation OpenGL) arrive en version 9. Elle supporte OpenGL 3.1 bien que le support matériel reste encore assez limité, en attendant de nouveaux pilotes. Parmi les nouveautés, on notera l'arrivée d'OpenCL, des pilotes Gallium 3D, VDPAU (support limité). Les développeurs annoncent des améliorations notables des performances même si un important focus a été réalisé sur le matériel.

» Java 8 se prépare lentement

mais sûrement (2013). Oracle a annoncé, avec AMD, l'incorporation du projet Sumatra. Le principe est de transformer du bytecode Java en code OpenCL quand une GPU compatible est détectée, ce qui est le cas de la plupart des GPU récentes. HotSpot JIT générera le code nécessaire pour OpenCL. Cela ne concerne pour le moment que Java mais d'autres langages supportés par la JVM pourront venir à l'avenir (ex. : Scala).

Site : <http://openjdk.java.net/projects/sumatra/>

» Un connecteur Mylyn / Tuleap Open ALM sera bientôt disponible dans la fondation Eclipse. Mylyn permet de gérer des tâches dans Eclipse. Il existe seulement 2 connecteurs Mylyn Task officiellement distribués par Eclipse : un pour Bugzilla et un pour Trac. Ce 3e plug-in offrira de nouvelles possibilités. Le développement est assuré par l'éditeur Obeo, membre d'Eclipse, avec les développeurs de Tuleap.

« Avec Tuleap, notre objectif est de faciliter le travail des équipes projets, notamment des développeurs. Ce nouveau connecteur Mylyn rentre dans cette stratégie. Plus on facilite le travail et automatise les développements, plus les équipes gagnent en efficacité. Nous souhaitons garantir à nos clients des développements de haute qualité. C'est pourquoi nous avons choisi de travailler avec Obeo, la société française qui a développé la plus forte expertise sur Eclipse » précise Manon MIDY, Responsable Marketing chez Enalean.

» Coverity annonce une nouvelle évolution de sa plateforme de tests de développement, suite intégrée pour tester rapidement et qualifier les projets. Plateforme de Test de Développement Coverity 6.5 incorpore des outils d'analyse et de couverture de codes. L'outil peut générer une cartographie du code et

du projet pour voir les dépendances et la couverture. Coverity fournit aussi aux développeurs des informations précises et exploitables pour réparer les défauts de qualité, de sécurité, les transgressions de test et les défauts d'analyse de tierces parties dans un workflow unique. La

DoppioVM

Welcome to Doppio! You may wish to try the following Java programs:

```
rhino
javac FileRead.java
javac Fib.java
java FileRead
java Fib <num>
java Chatterbot
java DiffPrint foo bar
java RegexpTestHarness
java Lzw c foo_lzw (compress)
java Lzw d foo_lzw foo (decompress)
```

The .java files can be inspected by typing 'edit [filename]'.

You can also upload your own files using the uploader above the top-right corner of the console.

The provided Java compiler is for Java 4, so programs using modern Java features (e.g. varargs) will not compile. However, you can still run them by compiling them locally and uploading their classfiles.

Enter 'help' for full a list of commands. Ctrl-D is EOF.

doppio >

» Doppio, un projet aussi original qu'ambitieux. L'objectif est d'exécuter du code Java sans aucun plug-in, en passant uniquement par une JVM écrite en JavaScript que l'on peut utiliser dans un navigateur. Il s'agit maintenant de proposer le maximum de classes et d'API Java dans le projet. Doppio permet déjà d'utiliser quelques bibliothèques de la JDK (string, pattern...), d'utiliser les génériques. Il reste un très gros travail à faire et notamment d'optimisation. Les dernières versions des principaux navigateurs sont compatibles. Pour voir les premiers résultats : <http://int3.github.com>



TELECHARGEZ LA VERSION
D'ESSAI GRATUITE
INFRAGISTICS.COM/DOWNLOADS

ETRE RAYONNANT

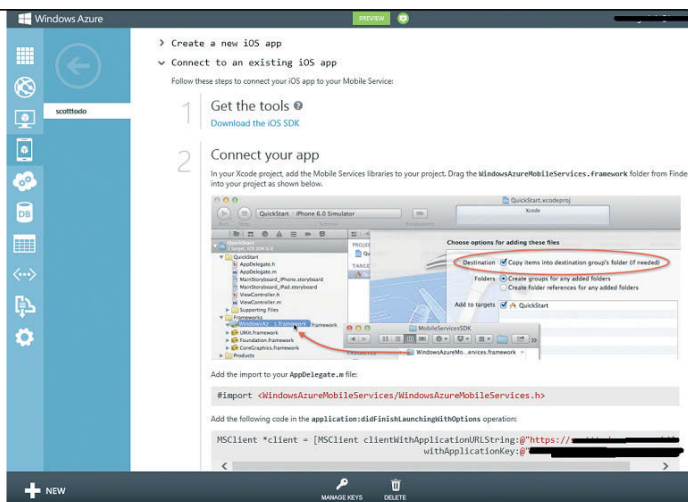
Avec des données éblouissantes

infragistics.com/ **EXPERIENCE**

INFRAGISTICS™
DESIGN / DEVELOP / EXPERIENCE

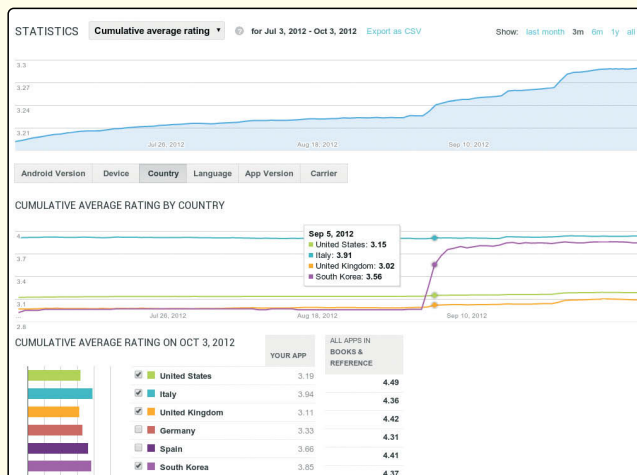
Infragistics Ventes France 0800 667 307 • Infragistics Ventes Europe +44 (0) 800 298 9055

Copyright 1996-2012 Infragistics, Inc. All rights reserved. Infragistics and NetAdvantage are registered trademarks of Infragistics, Inc. The Infragistics logo is a trademark of Infragistics, Inc.



» **Windows Azure Mobile Services** est maintenant compatible avec iOS, Facebook, Twitter, Google Identity, les mails, SMS, les blobs et bien d'autres choses ! Windows Azure Mobile Services facilite l'usage des services Azure dans les applications. Le service permet principalement de gérer et stocker les données, d'assurer l'identification des utilisateurs (et le single sign-on) et inclut un système de notification.

» **Google Play propose une nouvelle console** pour les développeurs. Il est ainsi possible de mieux voir le comportement de votre application sur la boutique Google Play. Les données (versions systèmes, les terminaux, la langue, le pays...) sont mieux présentées et plus complètes. La publication a subi elle aussi des améliorations tout comme la partie localisation.



» Des mises à jour mineures ont été dévoilées mi-octobre : **PHP 5.4.8** et **5.3.18**. Ces versions corrigent plus de 20 bugs et problèmes. Tous les développeurs et webmasters sont invités à déployer les nouvelles versions. Site : <http://www.php.net/downloads.php>

» L'éditeur **SoftFluent** annonce le **lancement de CodeFluent Entities** pour les applications Windows 8. Vous pouvez générer des Web Services pour mobile tout comme des applications Windows 8 complètes en quelques minutes ! Avec CodeFluent Entities, les développeurs peuvent s'affranchir du poids des habituelles montées de version technologiques et se concentrer sur le développement des nouvelles fonctionnalités nécessaires à leurs applications.

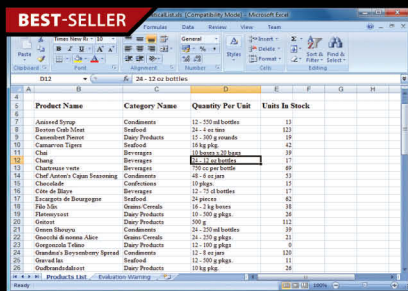
» **Webplatform.org**, un portail de référence pour le développement Web, voilà l'ambition du site qui est activement supporté par Microsoft, Google, Nokia, W3C, HP, Mozilla, Adobe et Apple. WebPlatform doit permettre aux développeurs web de trouver les ressources, informations, éléments techniques pour leur faciliter la vie et réduire les problèmes. On accède à de nombreuses documentations, des tutoriaux, un forum. Si le contenu est encore un peu limité, celui-ci augmentera semaine après semaine. Bref, une initiative à suivre !
site : <http://docs.webplatform.org>

» **NetBeans 7.3 est disponible** en préversion depuis quelques jours. Cette version aura une forte connotation web : développement d'applications HTML 5, éditeur JavaScript largement amélioré, nouvelles fonctions CSS, debugger JavaScript, nouvelles fonctions JavaFX, intégration de Groovy 2.0. Et comme toujours diverses améliorations pour les développements Java, Java EE et natif.
Site : <http://netbeans.org/community/releases/73/>

» **jQuery Mobile 1.2.0 est disponible**. Cette version introduit de nombreuses optimisations et améliorations. Un des gros changements est le widget popup pour produire rapidement des popup. On peut aussi créer des listes rétractables. Cette version supporte jQuery 1.8.2, iOS 6, Android 4.1, Tizen, Kindle Fire HD.

» Pets	
» Farm animals	
» Wild Animals	
Aardvark	»
Alligator	»
Ant	»
Bear	»
Beaver	»
Cougar	»
Dingo	»

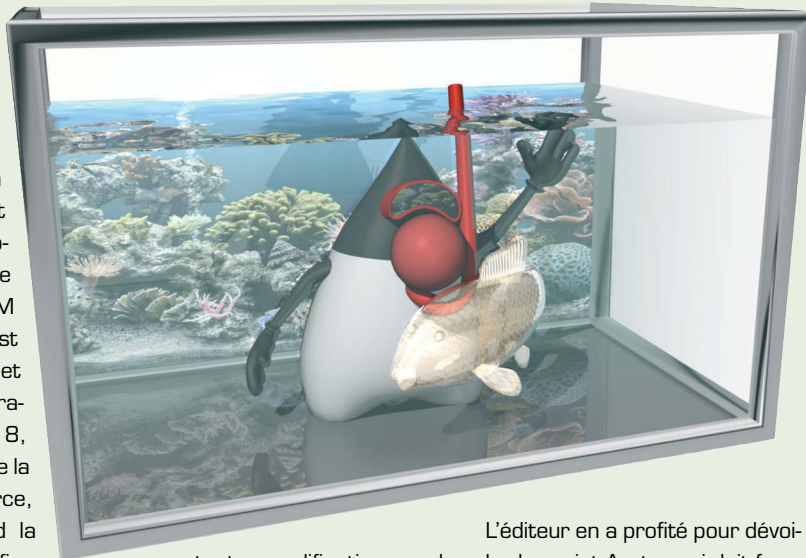
» **Microsoft et SuSe continuent à travailler ensemble**. Cette relation se renforce avec plusieurs annonces. Tout d'abord, la sortie d'un pilote (en bêta pour le moment) SQL Server pour permettre aux applications tournant sur SuSe Enterprise Server de se connecter à SQL Server. Ce pilote fonctionne avec Azure Database et SQL Server déployé en VM sur Azure. Côté virtualisation, de nouvelles améliorations ont été apportées sur le support de SuSe en contexte Hyper-V et aussi la livraison et la maintenance de code au noyau Linux, sur ces fonctions de virtualisation pour Microsoft. L'ensemble des serveurs Linux, quelle que soit la distribution, a ainsi accès à ce code source. Autre point important : l'administration. L'objectif est de simplifier l'administration SuSe depuis System Center, en supportant SuSe Manager. SuSe Studio n'est pas oublié. Car depuis le Studio il est possible de déployer une appliance dans le service IaaS d'Azure. Aujourd'hui, Microsoft est en pleine redéfinition de l'agenda open source et des objectifs futurs. OpenStack, Big Data sont sans doute au programme. Mais surtout, Microsoft est ouvert à d'autres distributions. Pourquoi pas un jour sur Azure, Debian et Fedora ? Dans un Cloud hétérogène, le support de multiples technologies et systèmes est incontournable. Et pour aider les utilisateurs et les clients, le centre d'interopérabilité de Paris, déploie les outils SuSe.



» Quel agenda pour Java ?

Après les polémiques sur le manque de réactivité de correction de bugs dans la plateforme, Oracle a réaffirmé son attachement pour Java et a, une nouvelle fois, dressé les grandes lignes du futur. Nous pouvons déjà retenir que Java 8 est maintenant prévu pour fin 2013, mais allégé de plusieurs fonctions, comme le retrait de Jigsaw devant apporter une plus grande modularité. Ce module sera, si tout va bien, inclus dans Java 9... Java 8 devrait être mieux adapté aux terminaux ayant peu de ressources. Mais Java 8 conserve plusieurs nouveautés importantes : closures, un nouveau moteur JavaScript, une nouvelle API Data / Heure,

le retrait de la génération permanente. Un des gros chantiers est la fusion entre la JVM HotSpot et JRockit. JDK 7 proposait une première version de cette JVM consolidée mais c'est avec la v8 que le projet va aboutir. Si Oracle travaille sur Java 8, OpenJDK s'occupe de la version open source, JDK 8, qui reprend la totalité des spécifications officielles. Sur JavaFX, notons tout d'abord qu'elle deviendra l'interface par défaut de Java embarqué. HTML 5 fait son apparition, avec d'impor-



tantes modifications sur la partie 3D, l'internationalisation, SceneBuilder. D'autres modifications interviendront plus tard. Mais JavaFX est plus que jamais un projet sensible pour Oracle.

L'éditeur en a profité pour dévoiler le projet Avatar qui doit fournir des applications HTML 5 et Java, sur toutes les plateformes mobiles. Attendons maintenant la réalisation de ce planning qui va encore bouger...

» Red Hat renforce JBoss.

L'éditeur continue à miser sur JBoss et particulièrement sur deux axes : SOA et BPM. Les technologies SOA et BPM ont mûri et atteint un niveau élevé de prise en charge au sein de secteurs d'activités critiques et dynamiques tels que les services financiers et la santé, où la



dynamique du marché est en constante évolution et les mandats de réglementation exigent des capacités d'intégration fluides pour relier les processus dans des scénarios de cloud hybride et d'itération rapide de règles métier. Ces nouveautés sont la suite logique des rachats de FuseSource et Polymita.

» **Mozilla rajoute à Firefox** de nouveaux éléments pour les développeurs : une nouvelle barre des tâches, qui permet de contrôler facilement, à partir de son

clavier, les outils de développement disponibles sur Firefox, en complétant les commandes et paramètres.

Firefox optimise certaines fonctionnalités telles que CSS3 Animations, Transitions, Transforms, Image Values, IndexedDB et Value & Units. Le navigateur optimise également les API Batterie et Vibration, deux API Web créées en partie par Mozilla. Firefox pour Android inclut désormais le mode « Lecteur » afin de faciliter la lecture et le partage d'articles à partir de votre mobile ou votre tablette.

Il suffit de cliquer sur l'icône « lecteur » dans la Awesome Bar pour l'activer. Cet outil vous permet de profiter pleinement de votre lecture en ligne sans interruptions (notamment publicitaires), avec des images au format adéquat et une police optimisée.

» Agilité : au-delà du développement ?

L'éditeur Seneca a mené l'enquête. Cela a donné lieu à la publication d'une étude complète il y a quelques semaines. Les trois principales conclusions sont :

- La démarche agile fonctionne bien mais il reste des progrès à faire : l'étude a montré que les pratiques de développement agile ne sont toujours pas largement adoptées ; 49 % déclarent que leur entreprise réalise

des travaux de déploiements agiles. Toutefois, parmi ceux ayant déclaré avoir effectué des déploiements, 52 % indiquent qu'ils sont satisfaits de leurs projets agiles actuels. L'étude a également montré que 55 % des participants étudient la possibilité de procéder à des travaux de livraison continue ou en réalisent déjà. La livraison continue étant devenue un sujet d'actualité.

- La communication est le principal défi à relever : selon les résultats de l'enquête, plus de 50% des participants ont indiqué que la compréhension et la priorisation des demandes des clients est le domaine qui requiert le plus d'améliorations pour assurer la réussite de la démarche agile. Les responsables et les développeurs ont convenu que 4 des 5 premiers obstacles concernaient la collaboration avec les autres équipes et les clients.
- Les parties prenantes doivent être plus impliquées dans les nouveaux projets de releases : alors que les équipes de développement agile réussissent bien, les participants à l'enquête estiment que les parties prenantes en amont et en aval sont exclues de la boucle de communication lors des sprints. Même s'ils constituent la première ligne de réponse, les membres de la Production IT et du Support sont impliqués dans moins de 40% des plans de releases.

Et si vous pensiez aussi à l'onduleur ?

Le mois dernier, nous avons évoqué la question de la carte graphique. Ce mois-ci, nous aborderons un sujet non moins important : l'onduleur. Ou comment éviter des coupures ou de griller un élément de son infrastructure.

Basiquement, l'onduleur, comme vous le savez, sert à éviter toute coupure dans le fonctionnement d'un matériel durant x minutes en cas de panne du réseau électrique. En fait, cette protection couvre plusieurs réalités : les prises parafoudre, les prises lissant le courant électrique pour éviter des sur ou sous-tensions. Il ne faut jamais oublier que le courant électrique n'est pas stable et que sa qualité varie. Même si vous n'utilisez pas d'onduleurs, il est toujours conseillé de s'équiper de prises de protection. La surtension peut provoquer la panne, la destruction de composants électroniques, voire d'un matériel dans son entier.

> Trois qualités d'onduleurs

Comme nous l'a précisé APC (Schneider Electric), il existe trois grandes catégories d'onduleurs :

Technologie	Usage
off-line	matériel non critique, poste de travail
line interactive	mixte entre le online et le offline
on-line	matériel critique, serveur critique

En général, les onduleurs de faible puissance et d'entrée de gamme seront Off-line. La principale différence entre le Off-line et le On-line est que en Off-line, le matériel branché sera « alimenté » continuellement par le réseau électrique et non par la batterie. La batterie prend la main uniquement en cas de coupure du réseau. Avec un temps de latence pour cette reprise. En On-line, le matériel est continuellement alimenté par les batteries de l'onduleur. Le line interactive compense une baisse de tension du réseau jusqu'à un certain niveau avant de passer sur la batterie, la gestion est plus dynamique que le Off-line et peut être un bon compromis pour un développeur. En Off-line et line interactive, le courant est filtré et lissé pour fournir une tension stable et de qualité (ouf, nous voilà rassurés !).

Un Off-line offrira globalement une autonomie (indice mi charge) de 10 à 15 minutes, 20-25 pour un line interactive et 30 pour un On-line (en « entrée de gamme »).

> Comment choisir ?

Nous vous conseillons, si votre budget le permet, d'opter pour line interactive, le must sera un On-line pour votre serveur et/ou vos unités de stockage et routeurs. Ensuite, il y a la puissance. Sur le courant, on parlera de Watts alors que sur un onduleur, la mesure se fait en VA (Volt Ampère). C'est idiot mais c'est comme ça. Il existe différentes manières de calculer le VA nécessaire par rapport aux Watts de son matériel, mais faisons simple : APC prend un coefficient de 1,66. Par exemple : 450 Watts = environ 750 VA (747 en réalité).

Vous devez additionner les Watts de vos différents matériels pour estimer le VA nécessaire. Mais attention, il est vivement conseillé de sur-dimensionner l'onduleur. Car si un matériel de 450W démarre, il va y avoir un pic de puissance nécessaire durant un court instant avec retombée à sa puissance nominale. Or, il faut que l'onduleur absorbe ce pic.

Outre le VA et le type d'onduleur, vérifiez aussi le nombre de prises connectées à la batterie. Car selon les gammes, si votre onduleur possède x prises, seulement une ou deux seront effectivement prises en char-



ge par la batterie en cas de coupure. Certains modèles (onduleurs haut de gamme) permettent de chaîner plusieurs onduleurs pour qu'ils fonctionnent en parallèle. Par exemple, le constructeur Eaton propose la technologie Powerwave Hot Sync. Cette redondance est idéale pour les configurations critiques.

> Administration

Les onduleurs sont livrés avec des consoles locales et acceptant un accès distant (selon le modèle et avec une adresse IP). On y trouvera les informations standards : niveau des batteries, état général de l'onduleur. Parfois, un écran LCD est disponible sur l'onduleur et fournit les données essentielles. Sur le haut de gamme, la console peut envoyer des notifications et erreurs. Parfois, vous pouvez définir des scripts de shutdown pour éteindre proprement et dans un ordre défini les serveurs et éviter tout problème d'intégration au redémarrage. Il vous faudra aussi surveiller la batterie. Plus elle sera sollicitée, moins elle va durer longtemps. APC estime à 3 ans la durée de vie optimale de la batterie. L'état de celle-ci influence directement le temps de charge et l'autonomie. Il ne restera plus qu'à les changer...

Merci à Olivier Bazin de Bezons (IT Distribution Sales Manager, APC by Schneider Electric).

François Tonic



» Les premiers modèles Windows 8 arrivent !

Alors que le système est sorti le 26 octobre dernier, Dell a annoncé la disponibilité pour la même date d'un ultrabook (XPS 12) à 1200 €, et d'un tout-en-un à 1300 €. Le XPS est un convertible avec écran basculable, design qui fleurit ici et là pour Windows 8, mais les tarifs restent assez élevés. Asus a dévoilé mi-octobre un ordre de prix pour les tablettes Windows RT : 599 \$ (modèle 10"1, 32 Go de stockage, HDMI, processeur Tegra 3) ! Là encore, le prix paraît un peu élevé par rapport à la concurrence. Premiers verdicts dans les prochaines semaines !

» Vous rêvez d'un clavier rétro éclairé pour votre PC ? Logitech propose le K810,



clavier sans fil Bluetooth, compatible Android, iOS, Windows, OS X... Il n'intègre pas de pavé numérique. L'autonomie annoncée est d'une dizaine de jours.

Prix : 99,99 €...

» Philips lance le Dictation Recorder for Android,

un enregistreur de dictées pour Android. Grâce à ce nouveau logiciel, les utilisateurs peuvent enregistrer, modifier et envoyer des fichiers de dictées à partir de leurs smartphones Android, ce qui leur donne plus de flexibilité et de mobilité, et leur permet de réduire les délais de traitement des documents. Les enregistrements

finalisés peuvent être envoyés en vue de leur transcription, puis retournés par e-mail à l'appareil Android de l'expéditeur. L'enregistreur de dictées Philips pour Android est également doté d'un large éventail de fonctions incluant le filtrage de fichiers audio, l'ajout de photos comme pièces jointes à des fichiers audio, le cryptage de fichiers 128 bits et la livraison de fichiers audio vers différents serveurs mobilesSpeechExec. Il offre une intégration

homogène au logiciel de flux de travail SpeechExec et prend en charge l'anglais, l'allemand, le hollandais, le français et l'espagnol. Fonctionne aussi sur iOS et Blackberry.

» PNY lance une nouvelle carte graphique : GeForce GTX 650 Ti.

Elle utilise la technologie NVidia Kepler, PureVideo HD, 3D Surround. Elle est optimisée par les derniers jeux du marché et notamment Assassin's (livré en édition limitée). La carte intègre 1 Go de ram, en adressage 128 bits, et 768 cœurs CUDA. Elle fonctionne bien entendu avec DirectX 11 et inclut le DVI, mini-HDMI (dommage !) mais pas de Thunderbolt. Prix public : 149 € TTC.

» Toshiba sort une nouvelle gamme de disques durs

pour les portables et les systèmes multimédia. Le MQ01ABF se caractérise par une faible consommation d'énergie, et présente les mêmes protections contre les chocs et les vibrations que son petit frère le MQ01ABD, un disque de 9,5 millimètres d'épaisseur, dont les capacités varient de 320 Go à 1 To. Ceci le rend parfait pour les gadgets mobiles, comme pour les portables compacts et les magnétoscopes conçus pour les espaces restreints. Pour le moment seuls les modèles 250, 320 et 500 Go sont disponibles.

» Gigabyte sort le P2542G. Le constructeur est spécialisé dans les machines portables pour les gamers. Et c'est comme toujours du très lourd : écran 15,1 pouces, GeForce GTX 660M avec 2 Go de ram, Core i7, Full HD 1080p, disque de 1 To ou SSD, démarrage ultra-rapide. Quelques fonctionnalités exclusives GIGABYTE ont aussi été ajoutées

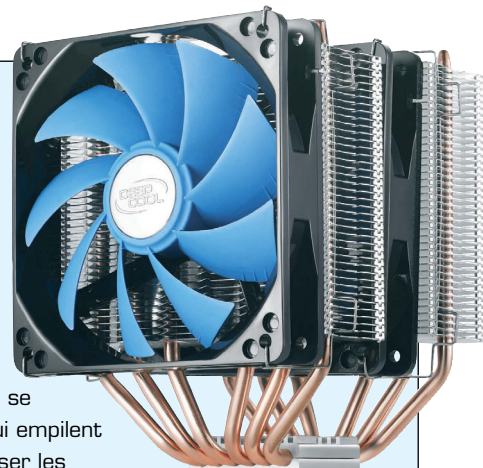
au cœur du P2542G comme un bouton de restauration système permettant de revenir à la configuration par défaut et un accès direct aux commandes multimédia et aux utilitaires en un seul clic. Le tarif est à la hauteur de la machine : 1529 €, avec disque dur et SSD. Par contre prévoyez un solide sac à dos : 2,6 kg !



» Deepcool sort un nouveau ventirad, NETPTWIN.

Le NEPTWIN est un système de refroidissement de haute qualité dont la base en cuivre est directement en contact avec le processeur, puis se sépare en deux tours qui empiètent 80 ailettes pour maximiser les performances de refroidissement.

Toutes les ailettes prennent la forme de vagues pour permettre une meilleure circulation de l'air. Il est proposé avec un tube de pâte thermique haute performance Z9 pour assurer un contact optimal. Il est aussi fourni avec tout le nécessaire pour être installé avec un processeur Intel et AMD, incluant les plaques arrière, celles en métal et les clips. Prix annoncé : 49,99 €



JS.everywhere(2012) Europe

JavaScript is ready for business • 16 - 17 novembre 2012 • Paris

JS.everywhere() est une conférence internationale entièrement consacrée à l'essor de JavaScript dans le secteur informatique professionnel : Web Apps, Mobiles Apps, JavaScript est désormais partout. Deux jours de rencontres et d'échanges avec plus de 300 experts et passionnés venus de toute l'Europe.

1ère édition européenne

Au programme

- Keynotes & sessions sur : Server-Side JavaScript, NoSQL, HTML5, Model Driven Architecture, sécurité, accessibilité, etc.
- Ateliers "hands on" : fondamentaux du développement Web et Mobile moderne
- Rencontres et networking, repas, goodies, au Tapis Rouge (Paris 10^{ème}), un lieu unique chargé d'histoire réservé pour toute la conférence
- En bonus, ateliers Wakanday : découvrez Wakanda, la nouvelle plateforme 100% JavaScript

Note : la majorité des contenus sera présentée en anglais

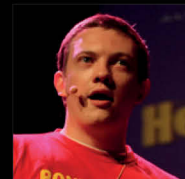
Speakers Confirmés



Douglas Crockford



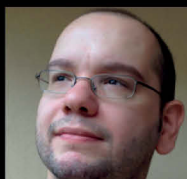
Laurent Ribardiére



Michal Budzynski



Estelle Weyl



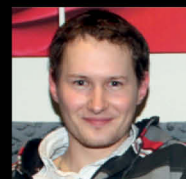
Dio Synodinos



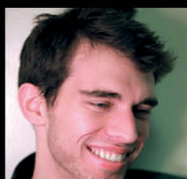
Alexandre Morgaut



Melinda Gallo



Vojta Jina



Domenic Denicola



Doris Chen



Ben Vinegar



Luca Garulli

Et bien d'autres à venir ...

Sponsors & Partenaires



Infos & Inscriptions
jseverywhere.eu

Un événement **4D** et **wakanda** avec le concours de

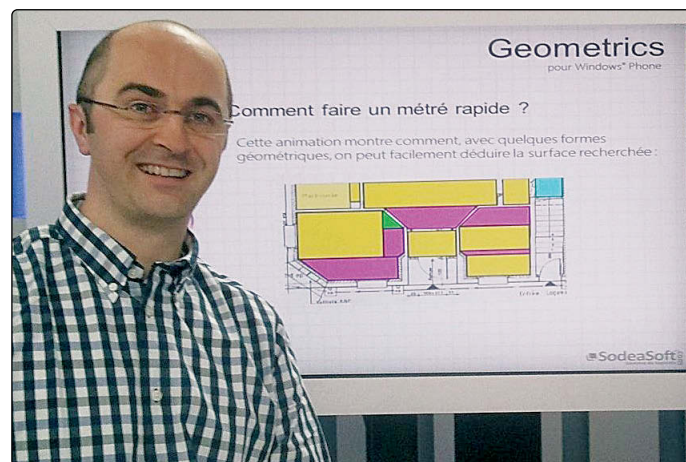


#jse2012



Du CAP au métier de développeur !

Christophe Peugnet, 39 ans, a commencé avec un CAP/BEP/BAC F3 (électro-technique). Il travaille ensuite comme métreur dans une entreprise du bâtiment. Puis pour valider ses compétences en informatique, il prend des cours du soir durant 3 ans au CNAM, ce qui lui permet d'obtenir un poste de développeur dans une petite entreprise de la région parisienne où il a travaillé deux ans pour enfin créer sa propre société en 2003, EBLM (<http://www.sodeasoft.com>), qui développe les logiciels SodeaSoft.



Comment es-tu tombé dans l'informatique et plus spécialement dans le développement ?

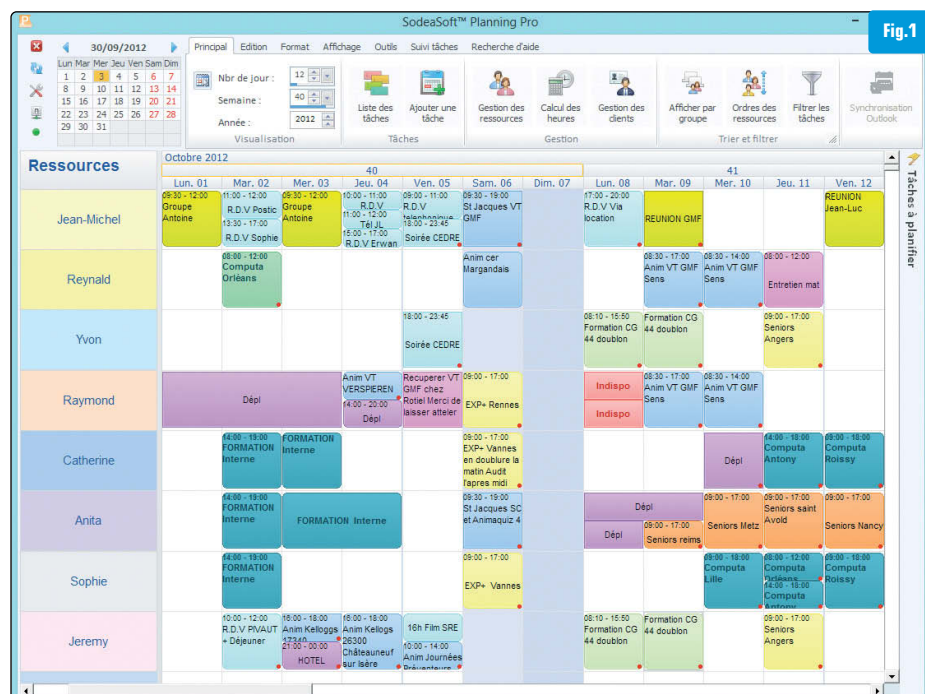
Christophe : Le premier clavier que j'ai touché c'était vers l'âge de 11/12 ans sur un Oric Atmos puis un Atari 800XL. C'est sur ce dernier que j'ai commencé à taper mes premières lignes en Basic. A l'époque il y avait le magazine Tilt qui proposait des dizaines et des dizaines de pages de code pour différents systèmes : mes premiers pas vers l'apprentissage du développement. Le côté frustrant, c'est que je ne pouvais pas enregistrer mes programmes car je n'ai eu un lecteur de cassettes que bien plus tard ! Il y avait la solution de laisser le micro-ordinateur allumé (mais cela monopolisait la

télévision avec la prise péritel) ou de tout retaper le lendemain. Parallèlement, à l'école arrivaient les micro-ordinateurs de la marque Thompson comme le M05, T07 mais aussi l'Apple IIc, mais à mon grand regret, je n'y ai pas appris grand-chose. Une petite anecdote qui me fait maintenant sourire c'est que des clubs d'informatique communaux se sont créés à l'époque où l'on apprenait aux adultes à coder en Basic... Peut-être voulait-on en faire des développeurs !? Dans les années qui ont suivi, j'ai dépensé toutes mes économies dans l'informatique, en marquant toutefois une pause dans le développement. J'ai tout d'abord fait l'acquisition de l'Atari 520 ST avec son lecteur de disquette 3'1/2 et sa souris. Mais

très vite je suis passé sur Amiga où je passais beaucoup de temps à composer de la musique (une très bonne machine pour nous aussi, NDLR) sur le célèbre Soundtracker, à rédiger et imprimer un fanzine sur les différents freewares et sharewares du moment. A la fin de vie des ordinateurs Commodore, c'était soit un PC soit un Mac... Et bien la roulette s'est bizarrement arrêtée sur le PC. C'est là que je me suis remis à coder en Turbo Pascal puis en Visual Basic 3.0. Vers 1995 j'ai commencé à diffuser sur internet mes premiers Sharewares, un jeu de cartes Réussite, un progiciel de devis de facturation, puis un logiciel de rentabilité et de planification pour la gestion du personnel [Fig.1]. Pendant des années, après le travail, j'ai codé des heures et des heures pour faire évoluer mes logiciels. Finalement, en 2003 j'ai créé ma société afin de commercialiser mes créations.

Pour toi, qu'est-ce qui fait que l'on aime toujours et encore le développement, la technique ?

Christophe : Sur mon premier Atari, quelle joie j'ai eu, après avoir mis des jours et des jours à trouver l'instruction SOUND, de pouvoir m'amuser à coder mon premier piano où avec mon Joystick (il n'y avait pas encore de souris), je cliquais sur les touches pour émettre des sons affreux. Et quel plaisir de taper l'instruction RUN et de voir le résultat de ce que l'on a créé ! Et bien un peu moins de trente ans après, j'ai toujours la même excitation dans mon travail de développement ! Ce travail intellectuel de recherche d'une solution d'architecture ou de performance d'un algorithme et le plaisir de voir le



résultat de ce que l'on a bâti. L'évolution constante des techniques, du matériel, avec par exemple l'arrivée de capteurs sur les ultrabooks, enrichit continuellement nos possibilités. Nous ne sommes pas prêts d'être à court d'imagination.

Tu as gardé un regard très geek : gadget, veille techno, c'est important pour ton job et ta passion ?

Christophe : Quand j'y pense maintenant, le regard geek naît avec la passion du développement. Cela me semble indissociablement lié au développeur, cela fait partie de notre culture. C'est très important d'avoir une veille technologique, d'être réactif à toutes les évolutions. Après, le partage de cette culture n'est pas toujours évident. Quand je vois des clients qui sont toujours avec leur Windows XP et qu'ils ne projettent même pas d'en changer, allez leur dire que vous utilisez depuis plusieurs mois Windows 8 ! Mais je suis optimiste, je pense que ce fossé va se rétrécir avec l'arrivée massive des tablettes et smartphones pour ce Noël 2012. C'est un phénomène de mode, même si ces utilisateurs ne connaissent pas encore le besoin, la tablette sera dans tous les foyers d'ici peu.

Etre développeur n'est pas toujours facile : pression, évolution constante, frustration des projets et des "chefs", c'est quoi pour toi d'être développeur aujourd'hui ? Le job a-t-il changé depuis tes débuts ?

Christophe : Oui, mon métier a évolué par rapport à ses débuts ! Pourquoi ? Et bien même si je trouve que développer en .NET est plus simple qu'il y a 15 ans en Visual Basic, on est confronté à une évolution très rapide des technologies depuis ces dernières années et pire, ces derniers mois. Regardez, personnellement j'ai toujours soigné à l'extrême mes logiciels. J'ai toujours pensé une nouvelle fonctionnalité avec cette devise : « La forme avant le fond » C'est-à-dire ce que verra l'utilisateur avant de penser comment le faire ; tapisser avant de monter les murs. C'est parfois difficile de faire simple. De manière générale, le design n'a jamais autant été mis en avant. Maintenant, on revient à l'essentiel de la fonction que l'on développe.

Dans une grande entreprise ce doit être assez sympa de partager le travail. Lorsque l'on est le seul développeur, il faut donc être tout à la fois. Maintenant il faut constamment apprendre, et depuis mars dernier je n'ai pas chômé de ce côté-là et ce n'est pas fini. Comme je l'ai dit, l'arrivée prochaine de tablettes, sera étroitement liée au Cloud, pour prendre exemple, le nouvel écosystème que Microsoft nous apporte va modifier la méthode de travail de l'utilisateur. La pression que j'ai, c'est moi qui me l'impose à apprendre beaucoup en peu de temps. Si le code semble plus simple ou plus rapide à produire, c'est une connaissance beaucoup plus large qu'il faut acquérir.

Et en dehors du boulot, qu'est-ce que tu aimes faire ? Comment trouves-tu l'équilibre entre travail, vie privée, passion, famille ?

Christophe : J'ai la chance d'avoir un métier où je vis ma passion, où je décide ce que je vais faire et quelle voie je vais prendre. Le revers de la médaille est que je n'ai pas de limite dans les heures de travail. Mon bureau est chez moi, il m'arrive de me lever à 3 h du matin pour travailler car j'ai trouvé une solution à un problème ou après le dîner de travailler quelques heures. Il faut donc juste trouver un équilibre avec la famille. A côté de cela, j'adore bricoler, m'occuper de mon jardin ou entreprendre des travaux.

Peux-tu nous présenter ton quotidien en quelques mots ?

Christophe : D'une manière générale, une journée type commence à 7h où je lis mon

courrier, entre 9h et 10h, on fait un point avec ma collaboratrice sur les domaines commercial, technique et assistance aux clients. Puis la journée sera très fractionnée entre la réalisation de devis, de factures, répondre au téléphone, faire de la formation aux clients, de l'administratif. Entre deux, je développe mon site internet, j'améliore mes logiciels et je finis ma journée vers 19h. Avoir son entreprise ne veut pas dire faire 100% de sa passion, loin de là ! Mais c'est une belle aventure.

Comment vois-tu ton job évoluer ?

Christophe : Développer n'étant pas mon métier mais ma passion, l'évolution qui me plairait serait d'allouer une place plus importante dans mon quotidien à la « recherche et développement ». L'arrivée de Store sur les PC sera donc pour moi une opportunité pour inverser la balance. Avec un Store, le développeur n'a jamais été aussi proche de l'utilisateur. Ce que je veux dire, c'est que la vitrine est la même pour tous. On retrouvera des applications faites par de grandes entreprises à côté d'applications faites par des auto-entrepreneurs. Windows 7 a été installé environ 800 millions de fois, ajoutez à cela les futures tablettes Windows 8. Je vous laisse calculer. Moi je veux bien toucher 0.005% des utilisateurs avec une petite application à 0.99€. Pas vous ?

Des conseils aux étudiants et dévs qui nous lisent ?

Christophe : Le conseil que je peux donner est celui que j'aurais voulu qu'on me donne il y a plusieurs années : Rejoindre sans réfléchir les communautés de développeurs que l'on trouve sur les réseaux sociaux. J'y ai rencontré des personnes extraordinaires qui accueillent les gens à bras ouverts et partagent leur savoir sans limite. Dans le même principe on commence aussi à faire la même chose et partager avec d'autres tout ce que l'on apprend. Je n'ai jamais autant appris en si peu de temps. Aussi si l'aventure Windows 8 ou Windows Phone vous intéresse, n'hésitez pas à vous inscrire aux programmes d'accompagnement nommés « les accélérateurs Windows 8 / Windows Phone ». Moi ça m'a changé ma vie.



Et mon bureau ?

Mon bureau, c'est un écran avec mes confrères de la communauté de développeurs, un autre pour gérer le quotidien et un troisième pour développer, et je travaille systématiquement avec de la musique !

Notre sélection de livres

JON DUCKETT HTML & CSS

Pearson

HTML 5 et CSS 3 demeurent des énigmes pour vous ? Et si vous vous y mettiez sérieusement avec cet ouvrage dédié aux développeurs et designers. L'auteur se penche sur l'ensemble des règles fondamentales du web : comment créer une image web, les liens relatifs, etc. Très didactique, avec de nombreux exemples, ce livre se prend en main et se lit rapidement. Une bonne référence pour débiter en douceur, et se dire que l'on peut faire beaucoup avec « simplement » les fondamentaux.

CORINNE SCHILLINGER Intégration web les bonnes pratiques



Eyrolles

Le terme intégration fait parfois peur et cette réalité ne doit pas être occultée sur les projets web. Maillon indispensable dans la création de sites, il rassemble tous les éléments pour monter les pages et livrer un site performant, accessible et utilisable. Cela nécessite tout autant de compétences techniques en HTML, CSS et JavaScript qu'une bonne appréhension de la qualité web (ergonomie, accessibilité, performance...). Ce livre se structure en plusieurs parties : préparer son projet, élaboration du socle HTML, le CSS. Une bonne lecture qui pourra aider à mieux comprendre le projet web et les différentes parties le composant.

COLLECTIF Le Registre sous Windows 8

éditions ENI

Avec l'arrivée de Windows 8, il faut se mettre à jour sur les différents aspects systèmes. Le Registre est toujours le point sensible à connaître. Il traite tout d'abord de l'architecture du Registre, de son organisation et de son fonctionnement. Les auteurs présentent ensuite des techniques avancées permettant son administration à distance ainsi que les outils disponibles à partir de l'invite de commandes (Reg.exe, Regini, etc.). Ils détaillent ensuite comment utiliser les dif-

férents langages de programmation afin d'administrer efficacement les ressources réseau en donnant des exemples de scripts utilisant les fonctionnalités WMI ou celles offertes par PowerShell.

ARNAUD FAQUE Google Android efficace



Eyrolles

Comment être réellement efficace avec Android 4 ? Voici un livre qui s'intéresse au côté pratique et explique comment utiliser au mieux son smartphone Android. Cet ouvrage tient compte de la version Google Android 4.0 (Ice Cream Sandwich). C'est tout un mode de vie que l'auteur partage avec ses lecteurs, y compris les questions de vie privée. Il explique comment exploiter les fondamentaux Linux d'un smartphone Google Android, et toutes les grandes applications connectées d'un téléphone ou d'une tablette Android sont décortiquées (mail, SMS, appels, contacts, géolocalisation, messagerie instantanée et Google+). Il insiste également sur les aspects cruciaux de synchronisation et d'archivage, bref, de contrôle de ses données.

COLLECTIF La république des réseaux



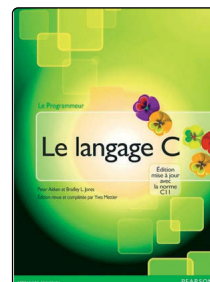
Fayard

On parle souvent de révolution numérique mais de quoi s'agit-il réellement ? Quels impacts sur la société, la vie sociale et l'économie ? Sommes-nous tous égaux devant le numérique ? Plus de deux milliards d'humains sont aujourd'hui connectés à Internet, faisant basculer dans le champ politique la question numérique, jusqu'ici cantonnée à la technique et à l'économie. Un livre qui fait réfléchir...

COLLECTIF Le langage C

Pearson

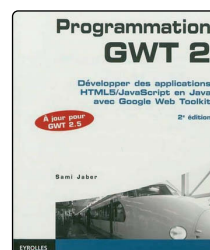
Le C reste le langage à apprendre, avant même de penser au C++ ou à tout autre lan-



gage objet. Grâce à des exercices pratiques et des cas concrets, cet ouvrage vous initie progressivement à toutes les bases du langage (fonctions, structures, pointeurs, gestion de mémoire,

gestion fichiers, bibliothèques de classes, etc.), vous apprend à utiliser les bonnes syntaxes et vous fournit de nombreux conseils, notamment en matière de sécurité. Cette édition supporte la toute nouvelle norme C11. Les exercices ont été réactualisés.

SAMI JABER Programmation GWT 2 2e édition



Eyrolles

GWT a donné un coup de fouet important sur la manière de concevoir et de développer des applications web. De la conception à l'optimisation et aux tests, toutes les étapes du développement sont déroulées, exemples de code à l'appui. S'adressant tant au développeur qu'à l'architecte, l'ouvrage dévoile les coulisses de chaque API au fil d'un cas d'utilisation simple et présente les nouveautés de GWT 2.5 que sont MVP avec Activity & Places, RequestFactory, CellWidgets et Editor.

STEFAN SCHARWK Android, apprendre à programmer des applis

Publitrone - Elektor



Tout est dans le titre : comment développer des applications mobiles pour Android ? Ce livre aborde l'ensemble des couches techniques nécessaires à la programmation : les outils, l'interface d'Android,

les concepts de base de Java, les applications web, la partie multimédia, Java, etc. Mieux vaut avoir des connaissances en Java pour utiliser ce livre et démarrer la programmation Android.

INVITATION

DÉVELOPPEUR - CHEF DE PROJET - WEBMASTER - DÉCIDEUR

ENEZ DÉCOUVRIR WINDEV®

PRÈS DE CHEZ VOUS

ET DÉCOUVREZ COMMENT DÉVELOPPER 10 FOIS PLUS VITE



918 nouveautés : Windows 8, Cube rOlap, graphe Surface, impression HTML, Ruban dans les éditeurs, FTP sécurisé, réplication multi-sites, CSS3, ...



Votre code est compatible :

- Windows 8 (7, XP, ...)
- Linux
- Mac
- Internet
- Intranet
- Windows Mobile & CE
- Windows Phone
- Android
- iPhone et iPad.



CALENDRIER

Montpellier	13 Nov
Toulouse	20 Nov
Bordeaux	21 Nov
Nantes	22 Nov
Bruxelles	27 Nov
Lille	28 Nov
Paris	29 Nov
Strasbourg	4 Déc
Lyon	5 Déc
Genève	6 Déc
Marseille	11 Déc

TOUR DE FRANCE
ENEZ DÉCOUVRIR
WINDEV 18
& WEBDEV 18
PRÈS DE CHEZ VOUS

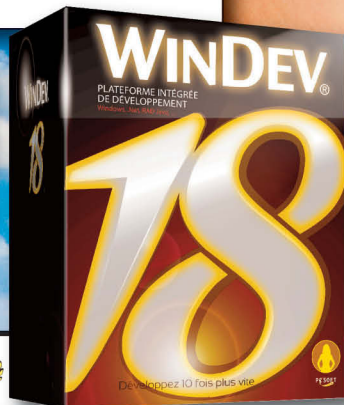
GRATUIT
inscrivez-vous sur www.pcsoft.fr
Attention: 10.000 places seulement

Elu «Langage le plus
productif du marché»

918
NOUVEAUTÉS

Fournisseur Officiel de la
Préparation Olympique

www.pcsoft.fr



» communauté

Duchess France : le JUG pour développeuses



Régulièrement, nous mettons en avant des développeurs ou des communautés qui gagnent à être connus. Ce mois-ci, nous vous proposons de découvrir l'association des Duchess, dans sa version française, qui s'adresse principalement aux (rares) femmes du secteur. De manière générale, en Occident, les « filles » sont relativement peu nombreuses dans les domaines techniques et plus spécifiquement dans l'IT et le développement. Elles ne représentent en moyenne que 8% des effectifs des écoles d'ingénieurs (ratio conservé dans l'industrie). Dans d'autres pays, comme en Asie, la parité est bien mieux respectée. Partant

de ce constat, des collectifs de femmes se sont constitués à travers le monde : Girls in tech (silicon Valley), Women 2.0 (orienté entrepreneuriat), Women who code (codeuses à San Francisco), etc.

« Un réseau mondial de développeuses Java... »

Le réseau international des Duchess, quant à lui, a la vocation de mettre en relation les femmes qui travaillent sur la plateforme Java. Le nom de l'association vient d'ailleurs de « Duke », la mascotte du langage. Duchess France a la particularité d'être un JUG (Java User Group) qui comptabilise plus de mille « followers » sur Twitter

(@duchessfr). Le réseau est présent à Paris (principalement) mais également en région : Limoges, Lyon...

Pour les Duchess France, la faible proportion de femmes dans les écoles d'informatique correspond à une désaffection pour ce type d'études et non à un simple rejet. Un de leur objectif a donc pour ambition de donner de la visibilité aux femmes du secteur, de les mettre en avant et de montrer aux plus jeunes qu'une carrière est possible. C'est essentiellement une mission de communication. Pour cela, les Duchess organisent régulièrement des événements (gratuits) ouverts à tous :

- des rencontres « open spaces », ou « Marmite », lors desquelles les participants et orateurs peuvent débattre d'un thème mensuel ;
- des ateliers « Hands on » permettant à de petits groupes de découvrir une technologie en mettant les mains dans le cambouis ;
- des soirées spéciales pour l'anniversaire du JUG avec un Trivial Java ;
- des annonces rapides « lightning talks » d'un quart d'heure

environ, en introduction de présentations ;

- des BOF (Birds Of Feather) comme celui de Devbox (Programmez N°153 de juin 2012).

« Les hommes sont également les bienvenus... »

En outre, un blog très actif propose des interviews, des retours de conférences et tient à jour un calendrier des manifestations. Au-delà de l'information, l'objectif est d'amener des femmes à se faire connaître par des articles techniques, en étant accompagnées, sur un site ayant déjà une audience forte.

Duchess soutient également d'autres communautés, notamment dans le cadre de la formation des jeunes, et peut intervenir pour présenter les métiers de l'informatique auprès d'écoles ou organismes.

Blog des Duchess France :

<http://www.duchess-france.org/>

Twitter des Duchess France :

@duchessfr



Thierry

Leriche-Dessirier

Twitter :

@thierryleriche

agenda

NOVEMBRE

- 01 novembre au 04 novembre, Paris Porte de Versailles, **Games week 2012**, 3e édition du rendez-vous, désormais incontournable du jeu vidéo en France. <http://www.parisgamesweek.com/>
- 9 novembre : **Drupagora** (Paris). Deuxième édition du principal événement français autour de Drupal : Drupagora. L'événement est orienté entreprise, chef de projet, DSI. Le développement sera aussi abordé. A ne pas rater : responsive design web, Drupal dans le cloud, les clés de la réussite d'un projet Drupal, présentation de Drupal 8... Site : <http://www.drupagora.com>
- 16 et 17 novembre : **JS.everywhere** (O 2012) (Paris). JavaScript est un mystère pour vous ? JS.everywhere est l'événement européen pour comprendre, approfondir et coder en JavaScript. De nombreuses sessions techniques seront présentées sur les deux jours et sur tous les sujets (données, interface,



HTML 5, debug, les outils, etc.). site : <http://jswhere.eu>

- **Tour de France PC SOFT** - Lancement de WINDEV & WEBDEV 18. Du 13 Novembre (Montpellier) au 11 décembre (Marseille) . Paris : 29 novembre. www.pcsoft.fr



- **Séminaire Magic Software France**, Paris Pavillon Kléber, Mardi 13 Novembre 2012 - 8h30 14h00. Vos Applications métier, prêtes pour la Mobilité. Magic Software vous dévoile tout : Nouvelle Image, Mobilité et Roadmap 2013 ! http://web.magicsoftware.com/PRMobilitySeminar13Nov12Paris_registration.html



- Les 29 et 30 novembre, CCO Nantes, 2e édition du **PHP Tour**, organisé par l'AFUP, Il s'adresse essen-

tiellement aux développeurs, utilisateurs, prestataires et passionnés de PHP.

<http://afup.org/pages/phptournantes2012/index.php>

ETRANGER

- Du 05 au 07 novembre 2012, Hôtel Fira Palace – Barcelone, **LinuxCon Europe 2012**. LinuxCon Europe, qui a fait ses débuts devant une foule à guichets fermés à Prague en 2011, Tient sa deuxième édition en Espagne. <https://events.linuxfoundation.org/events/linuxcon-europe>
- Du 09 au 11 novembre, Londres, Mozilla Festival 2012. Plus de 800 personnes venant de 40 pays différents y sont attendues. Codeurs, designers, cinéastes, joueurs viendront y partager leur expérience. <http://mozillafestival.org/>
- **Europe - 12-14 Novembre**, Café Moskau, Berlin, **Qt Developer Days 2012** L'événement s'étend sur trois jours et comprend une journée complète de formation, des pistes techniques, des études de cas de clients et beaucoup de temps pour le réseautage. <http://www.qtdeveloperdays.com/>

APPEL ! APPEL ! APPEL !

Vous êtes une communauté, un groupe utilisateur et vous voulez annoncer un événement, faire connaître votre groupe ?

Programmez ! est là pour vous. Envoyez-nous vos agendas, résumés, photos. redaction@programmez.com ou ftonic@programmez.com

Pourquoi choisir WinDev ?

La réduction des coûts reste l'une des principales préoccupations des DSI et des entreprises, quel que soit le secteur d'activité ou la taille de l'entreprise. D'autre part, de plus en plus de sociétés externalisent en tout ou partie leur DSI, et que ce soit le cas ou non, les DSI doivent rendre des comptes et ne gèrent plus seules leurs budgets. La refacturation interne est monnaie courante.

Windev permet une réduction de ces coûts et une réactivité de mise en œuvre et déploiement. Souvent, les coûts de réalisation sont plus élevés avec d'autres solutions (même à niveau d'expertise et d'expérience professionnelle similaires) ; soit à cause de la complexité du développement, soit à cause des infrastructures techniques nécessaires.

Voici quelques abaques expérimentés qui globalisent des coûts de développements (flux de données, IHM, règles de gestion et reporting) : pour réaliser une même fonctionnalité il faudrait 760 minutes en JEE, 630 minutes en .net et 480 minutes en WinDev. WinDev peut être utilisé en informatique de gestion ou industrielle, que ce soit pour des applications mono-poste, en réseau ou en client serveur. Pour des applications de taille moyenne, qui doivent être partagées entre plusieurs développeurs, l'utilisation du GDS (gestionnaire de sources) est une solution de partage, d'historisation et de sauvegarde. Il utilise une base de sources (procédures, classes, fenêtres, états, composants, analyses...) qui peut être installée sur un serveur ou un poste réseau.

Pour des grosses applications, l'utilisation des composants peut être une solution de découpage (par fonctionnalité métier confiée à un ou plusieurs développeurs par exemple), mixée ou non avec l'utilisation du GDS.

Note : un composant est un ensemble d'éléments qui effectue une fonctionnalité précise et qui peut être redistribué à d'autres développeurs. Hormis les coûts liés au temps de développement, nous parlerons de la formation qui favorise également WinDev dans le choix d'une solution. La prise en main de WinDev se fait de façon aisée.

Enfin, cet article concerne WinDev mais PC Soft propose 2 autres produits complémentaires que sont WinDev mobile et WebDev. WinDev mobile permet le développement d'applications mobiles pour smartphones, tablettes, Pocket PC pour Android, iOS, Windows phone... et WebDev est orienté applications Web et sites Internet.

> Quelques exemples d'applications réalisées avec WinDev (de 50 à plus de 10 000 jours hommes)

- Dossier Technique Produit (~ 40 jours) : Application réseau connectée via ODBC à Access, réalisée en 3 langues. Cahier des charges et spécifications techniques et nutritionnelles de produits industriels.
- Gestion industrielle du colostrum bovin (~250 jours) : Application client serveur connectée en accès natif à Oracle. Gestion des producteurs, de la collecte, des stocks et transports, de la fabrication et de la paie.
- Gestion des achats et de la logistique d'une entreprise de VAD (> 10 000 jours avec les maintenances évolutives). Application client serveur

connectée via ODBC à SQL Serveur (réalisée en partie en régie).

- Gestion de restaurant : Application réseau connectée au SGBD HyperFileSQL et communiquant avec une application WinDev mobile sous plateforme Android et SQLite.

Gestion des clients et fournisseurs, des achats et des ventes, des réservations et campagnes emailings, du stock, de la facturation et des livres de caisse et des recettes.

> Développer un ERP avec WinDev ?

En plus de mes missions de consultant ou de formateur, je réalise également des applications en tant qu'éditeur. Plutôt que de parler d'ERP, je parlerai de PGI (Progiciel de Gestion Intégré). Sortira en début 2013 un PGI destiné aux PME et TPE.

Ce progiciel, entièrement réalisé avec WinDev, HyperFile SQL et WinDev mobile englobera toutes les composantes nécessaires au fonctionnement d'une PME (front office, back office, gestion commerciale, finances, suivi de l'activité des collaborateurs, plannings, messagerie...).

> Comment utiliser WinDev ?

Avant de refondre un SI complet avec WinDev, il est utile de le faire prendre en main par les équipes.

Se documenter

Pour connaître les possibilités et les coûts :

- par la documentation mise sur le site de PC-Soft (éditeur de WinDev),
- par la participation aux « Tour de France » proposés par PC-Soft (séminaire technique de présentation)
- par la lecture d'ouvrages sur WinDev notamment les fondamentaux du développement avec WinDev (paru aux éditions ENI).

Se former

Des formations sont organisées par PC-Soft et un cours d'autoformation est fourni avec WinDev. De nombreux exemples complets ou didactiques sont également disponibles. Des formations sont dispensées par des organismes ou des consultants indépendants.

Les formations que je donne sont orientées en fonction de la demande du client sur des points précis (exemple : les meilleures pratiques de programmation avec WinDev et Oracle) et/ou orientées pratique (la création d'une application avec WinDev de A à Z).

Le fait que WinDev soit un AGL dont les éditeurs et outils sont intégrés permet d'orienter une formation par la pratique en abordant les éditeurs dans un ordre logique (éditeur d'analyse, éditeur de fenêtres, éditeur de code, éditeur de requêtes, éditeur d'états, éditeur d'installation...). Rien de tel que la pratique pour être au plus vite opérationnel.

Concrétiser

La prise en main par la réalisation sur le SI de l'entreprise doit être faite dans la continuité de la formation. Comment ? Par la création d'une application pilote permettant d'aborder des points importants comme la connexion au SGBD de l'entreprise.

Par le développement de fonctionnalités métier « basiques » qui sont les prémisses de la refonte du SI. Exemple : la gestion des données fournisseurs contenant une fenêtre, quelques champs (nom, adresse etc.) et quelques boutons (enregistrer, supprimer, imprimer...). Simple mais efficace car permettant de remettre en musique les éditeurs principaux et le W-Langage.

Et pour les développeurs un peu plus avertis : par la création de classes réutilisables (connexion et exécution des requêtes) ou de composants réutilisables (gestion des paramètres généraux ou des utilisateurs et habilitations).

> L'apprentissage

La facilité d'apprentissage de WinDev est un avantage majeur. En tant que consultant, je donne diverses formations sur WinDev. Les exemples suivants démontrent la facilité d'apprentissage de cet outil. Au pôle formation de la chambre de commerce et d'industrie de Colmar je donne une formation WinDev depuis plusieurs années. Cette formation, d'une durée de 5 jours, est destinée aux stagiaires en BTS Informatique de Gestion.

Ces stagiaires n'ont en général jamais pratiqué WinDev, et, très souvent, n'ont comme seule expérience informatique que les cours dispensés les semaines précédentes.

En 5 jours, ils sont capable de réaliser une application complète avec WinDev et ont eu l'occasion de mettre en pratique les étapes majeures du développement d'une application : le modèle conceptuel de données, le modèle logique de données, la génération de l'analyse, la création des IHM's et l'utilisation des champs les plus courants (tables, listes, combos, boutons, sélecteurs, interrupteurs...), le W-Langage qui permet de manipuler tous ces éléments, la construction des requêtes, la réalisation d'états, la compilation de l'application, la création d'un exécutable autonome et le déploiement. Le W-Langage, utilisé en français, est très intuitif et l'aide en ligne, qui est riche et illustrée, permettent d'être très vite autonome.

Autre exemple d'apprentissage : en entreprise, pour des informaticiens ayant de l'expérience mais ne connaissant pas WinDev, une formation de 2 jours leur permettra une autonomie suffisante pour développer leurs applications. Enfin, en coaching individuel, une journée intensive permet d'obtenir sensiblement le même résultat.

Il est évident qu'il n'est pas possible, en deux jours de formation ou en une journée individuelle de coaching, de voir toutes les fonctionnalités de WinDev, notamment les nombreux outils et éditeurs. Mais les essentiels nécessaires à la réalisation d'une application complète sont maîtrisables.

> Le temps de développement

WinDev permet de réaliser des applications très rapidement grâce à ses éditeurs. Je citerai ici le RAD (Rapid Application Development) qui permet de créer en quelques clics une application complète. Il se base sur des modèles de programmation préexistants qui permettent de créer l'application via un assistant. Existe également le RID (Rapid graphical Interface Design) qui permet de créer des fenêtres à partir de l'analyse liée au projet. Contrairement au RAD, avec le RID, le code nécessaire au fonctionnement des fenêtres reste à

développer. Ces 2 méthodes permettent de créer des applications en un temps record.

Néanmoins, lorsque le code est auto-généré, il est moins évident à modifier, puisque la logique n'a pas été pensée par le développeur mais par l'AGL.

J'ai expérimenté le RAD, notamment la création d'une application en moins de 5 minutes (de la création du projet à la génération de l'exécutable) pour l'écriture de l'ouvrage « WinDev 15 : Les fondamentaux du développement avec WinDev », paru aux éditions ENI en 2010.

Note : actuellement la V17 est sur le marché avec sortie de la V18 prévue prochainement.

Cela étant, même sans passer par le RAD ni le RID, le développement d'applications avec WinDev se fait de façon très rapide grâce à la puissance des fonctions et au nombre de fonctions du W-Langage. La liste est riche (environ 3000), mais au final on n'utilise que très peu de fonctions. Je pratique WinDev en entreprise depuis une quinzaine d'années, je n'ai pas vu de RAD implémenté dans les sociétés pour lesquelles je suis intervenu, et il ne m'a jamais été demandé de développer avec le RAD. Que ce soit des DSI de moins de 10 personnes ou de plus de 100 personnes, le développement des fonctionnalités a toujours été confié à l'informaticien.

De même pour mes applications réalisées en tant qu'éditeur, je n'utilise pas le RAD, je me sers beaucoup des fonctions et propriétés et utilise des classes, entre autres, pour les requêtes, ce qui permet de passer très facilement une application d'un SGBD à un autre.

L'extrait de code ci-dessous permet la connexion à un serveur FTP et l'envoi d'un fichier dans le répertoire sur ce serveur FTP (en bleu les fonctions) :

```
nConnexion est un entier
sLf_Serveur est une chaîne
sLf_User est une chaîne
sLf_mdpp est une chaîne

//Connecte le poste en cours à un serveur FTP (File Transfer Protocol).
nConnexion = FTPConnecte(sLf_Serveur,sLf_User,sLf_mdpp)
SI nConnexion = -1 ALORS
    Erreur("Erreur de connexion au serveur FTP:" + ErreurInfo())
    RETOUR
FIN

//Transfère un fichier ou un répertoire sur un serveur FTP (File Transfer Protocol).
SI FTPEnvoie(nConnexion, "MesDocuments\Fichier.txt", "/Temp") = Faux ALORS
    Erreur("Envoi du fichier impossible:" + ErreurInfo())
FIN
```

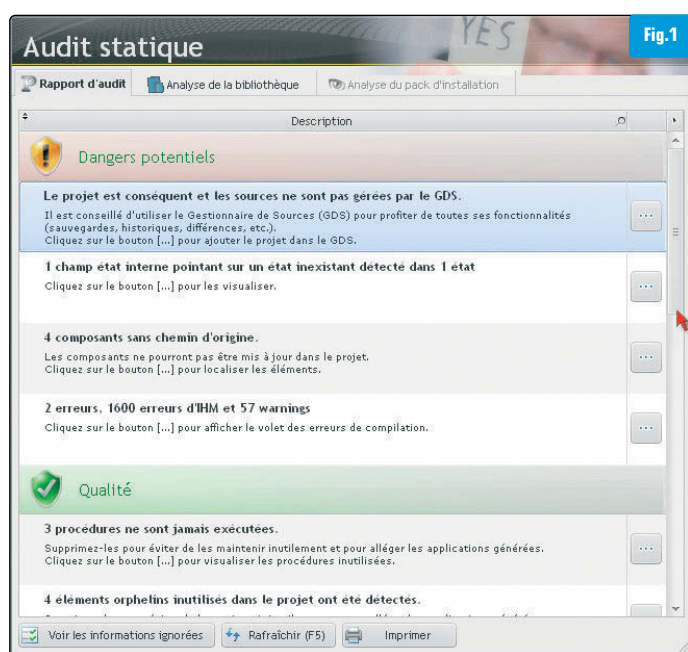
La fiabilité des applications et les migrations

Les applications sont fiables et robustes. Je prends l'exemple d'une application réalisée pour un département logistique, à laquelle j'ai participé en grande partie et que je maintiens encore actuellement. Elle contient environ 240 fenêtres, une vingtaine de classes et environ 140 000 lignes de code. Elle est connectée à une base SQL Serveur via ODBC et déployée sur une centaine de postes utilisateurs sur différents sites. Cette application a été initialement réalisée avec WinDev 5.5 ! Elle a évolué fonctionnellement et migré plusieurs fois en sautant de version de WinDev (de 5.5 à 7, de 7 à 14 notamment). Elle est stable depuis plus de 10 ans et n'a pratiquement jamais connu d'erreurs d'exception générale. La migration de version est facilement réalisable et prend peu de temps.

Voici l'exemple d'une application développée en V15 et migrée en

V17 récemment (elle contient 60 écrans et 12 000 lignes de code et est en cours de réalisation, ce qui explique les erreurs signalées par l'audit) :

- **11h15** : WinDev détecte que le projet a été créé avec une version antérieure et propose une sauvegarde et une mise à niveau du projet.
- **11h16** : proposition d'intégration automatique des nouveautés (aperçu avant impression au gabarit du projet, erreurs de compilation d'IHM, animations sur les champs, autorisation ou non par l'utilisateur du logiciel Etats et Requêtes, amélioration des IHM avec intégration des nouveautés, lancement de l'audit statique)
- **11h21** : l'audit statique donne son rapport : beaucoup d'erreurs d'IHM dues aux traductions non faites dans une ou plusieurs langues, des warnings dus majoritairement à des variables non utilisées et 2 erreurs bloquantes...
- **11h25** : les 2 erreurs bloquantes sont corrigées et l'application est disponible en V17 pour continuer les développements [Fig.1].



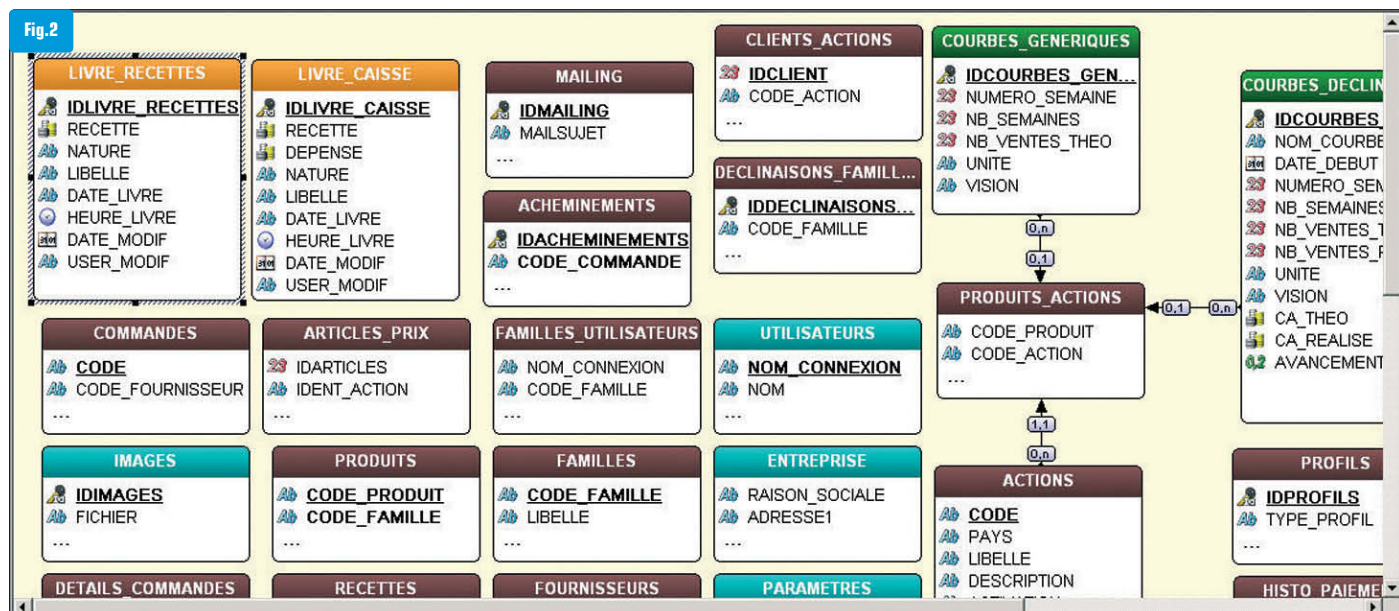
Rapport de l'audit statique lors d'une migration

Le coût

Le coût d'une licence WinDev est de 1650 € HT lors de la première acquisition. La mise à jour, actuellement annuelle, car PC Soft sort une version par an depuis quelques années, est de 590 € HT (tarif catalogue).

Compte tenu des éléments suivants, j'estime que le coût reste raisonnable :

- SGBD intégré (HyperFile SQL en mode réseau et en mode client/serveur) [Fig.2]
- accès aux bases de données tierces (SQL Serveur, Oracle...) via ODBC, OLE DB ou nativement. [Fig.3]
- nombreux outils intégrés (exemples : convertisseur de fichiers externes vers des fichiers HyperFileSQL, visualiseur de contenu d'un fichier HyperFileSQL, compresseur de fichiers sous forme d'archives, enregistreur et exécuter de scripts de test d'applications ...) [Fig.4]
- retour sur investissement rapide compte tenu des facilités de développements
- liberté de déploiement des applications et l'utilisation de HyperFileSQL permet également un déploiement libre des bases de données.



- gestion de projet intégrée

Cependant, pour une entreprise qui utiliserait également WebDev et WinDev mobile, ces coûts sont multipliés par 3.

Citons également les points suivants :

- la possibilité d'achats en licences multiples par 3, 5, 10 et 50 licences
- l'échange concurrentiel : 990 € HT au lieu de 1 650 € HT en joignant une preuve de possession d'un langage ou d'une base de données
- certaines offres promotionnelles du style : 1 PC pour 1 euro de plus...
- certains accès natifs qui doivent être acquis séparément et qui sont payants : exemple accès natif SQL Server 7 et supérieur

Note sur l'accès natif : celui-ci permet d'accéder à une base de données sans drivers externes depuis un programme en WLangage et permet notamment d'utiliser les fonctions HLitxxx

- l'existence d'une version Express (notamment de WinDev 17) : version gratuite limitée qui a pour but de découvrir les principales fonctionnalités de WinDev (pour les étudiants par exemple...). L'objectif n'est pas de développer des applications complètes avec une version d'essai.

[Fig.5] : Code de connexion à une BdD SQL Serveur via ODBC et exécution d'une requête avec récupération du résultat

En conclusion, pour répondre à la question : « Une société peut-elle créer ou faire créer toutes ses applications avec Windev ? » Pour des applications de gestion Windev est sans conteste une excellente solution, voire une des meilleures. J'ajouterai que, pour la mobilité des applications, WinDev mobile est un bon complément.

David Vandevilde - Consultant / Gérant
www.laciteduweb.com - www.expert-winddev.net



Fig.4



Fig.5

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

VITE, ABONNEZ-VOUS !

jusqu'à -50%

Code, gestion de projets, développement web, mobilité, Programmez ! est à la fois votre outil pratique, des articles de code par les meilleurs experts et votre veille technologique.



1

Abonnement 1 an au magazine

49 € (au lieu de 65,45 €, prix au numéro)

2

Abonnement Intégral : 1 an au magazine + Archives Internet et PDF 59 €

3

Abonnement 2 ans au magazine

78 € (au lieu de 130,90 €, prix au numéro)

4

Abonnement intégral 2 ans au magazine + Archives Internet et PDF 88 €

Toutes les offres en ligne : www.programmez.com

Abonnez-vous à partir de 3,80 € seulement par mois

Oui, je m'abonne

à retourner avec votre règlement à
Groupe GLI, 17 route des Boulangers 78926 Yvelines cedex 9

- ☐ **Abonnement 1 an au magazine** : 49 € (au lieu de 65,45 €, prix au numéro)
- ☐ **Abonnement Intégral : 1 an au magazine + archives Internet et PDF** : 59 € (au lieu de 65,45 €, prix au numéro)
- ☐ **Abonnement 2 ans au magazine** : 78 € (au lieu de 130,90 €, prix au numéro)
- ☐ **Abonnement intégral 2 ans au magazine + archives Internet et PDF** : 88 €

☐ M. ☐ Mme ☐ Mlle Entreprise : _____ Fonction : _____

Prénom : _____ Nom : _____

Adresse : _____

Code postal : _____ Ville : _____

Tél : _____

(Attention, e-mail indispensable pour les archives sur internet)

E-mail : _____ @ _____

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



Démystification des QR codes

Que ce soit sur les affiches publicitaires, les bouteilles d'eau minérale, les cartes de visite, dans les arrêts de bus ou même sur les pages de ce magazine où ils renvoient vers des contenus complémentaires, les QR codes ont envahi (timidement) notre quotidien.

En première approche, les QR codes ressemblent à des carrés composés de pixels blancs et noirs, qu'on appelle « modules », dont la quantité et la répartition représentent le message contenu dans le code. C'est en quelque sorte une « évolution » des classiques codes-barres (en une dimension) et des codes Data Matrix (deux dimensions), notamment présents sur les emballages des médicaments.



QR code contenant le message
« Le portail des développeurs »

> Un peu d'histoire

Initialement, les QR codes (pour Quick Response) ont été créés en 1994 par Denso-Wave pour le compte de la société Toyota. Cette dernière avait besoin d'un support simple, petit et puissant pour suivre les pièces dans ses usines automobiles. Le standard AIM (Automatic Identification and Mobility) a été publié en 1997. Deux ans plus tard, l'entreprise Denso a rendu libre l'utilisation du QR code et a publié le standard japonais JIS X0 510. Dans les années qui suivent, et jusqu'en 2009, plusieurs normes ISO (ISO/CEI 18004) viennent encadrer les spécifications du QR code et ses évolutions.

> Utilisations possibles

L'utilisation la plus courante des QR codes est sans doute d'envoyer l'utilisateur vers un site Web. Certains codes renvoient plus spécifiquement vers une page précise du site, correspondant par exemple à la description d'un produit commercial. On retrouve de plus en plus cette pratique dans les catalogues des supermarchés. Ainsi, lorsqu'on scanne le code associé à un produit, on est redirigé vers la liste des caractéristiques, avec des photos additionnelles, des avis d'utilisateurs, des codes promotionnels, etc.

Les offices du tourisme, les sites nationaux mais aussi les musées proposent également de scanner des QR codes. Cela permet d'effectuer une visite personnalisée, enrichie de nombreux contenus (Web, audio, vidéo, etc.), plus agréable qu'avec les anciens guides audio. En arrivant devant un tableau ou une sculpture, on obtient des informations sur l'œuvre, sur son auteur, des anecdotes, etc. Les possibilités sont nombreuses.

“ Une visite enrichie de contenus... ”

De façon un peu détournée mais très appréciée, de nombreux arrêts de bus proposent désormais de scanner des QR codes pour connaître les horaires de la ligne et l'heure de passage du prochain bus. Cette dernière est réactualisée en permanence en fonction de la position du véhicule, du trafic, des statistiques de circulation, etc. Pour cela, le QR code renvoie simplement vers une URL dont un des paramètres correspond à l'identifiant de la station. Cette URL est fixe mais le contenu de la page, lui, est dynamique.

Les QR codes contiennent simplement un message. Ce qui différencie les différents types de code, c'est le contenu du message, et plus

précisément sa structure. Ainsi, certains QR codes contiennent des cartes de visite au format meCard ou vCard (cf. signature de cet article), des paramètres de connexion automatique à des réseaux WIFI, des numéros de téléphone qui provoquent un appel lors du scan ou l'envoi d'un SMS, des dates d'événements (salon, concert, match de foot...) automatiquement ajoutés dans l'agenda, etc.

Pour réaliser l'une ou l'autre de ces actions, le message doit contenir des marqueurs spécifiques. Par exemple, pour lancer un appel, il faut utiliser le marqueur « tel ». Pour un email, on utilisera « mailto » comme on le ferait en HTML :

```
String email = «jean.dupont@hotmail.com»;  
String sujet = «Vacances»;  
String texte = «Bonjour%20bla%20bla»;  
  
String message = «mailto:» + email + «?subject=» + sujet + «&body=» + texte;
```

Il est possible d'utiliser des marqueurs propriétaires. La seule limitation est que le scanner sache les reconnaître. Ainsi, certaines sociétés de transport (train, avion, etc.) impriment des QR codes sur les tickets mais on ne peut les décoder que si on dispose du bon logiciel ; le contrôleur en est évidemment équipé.

> Structure des QR codes

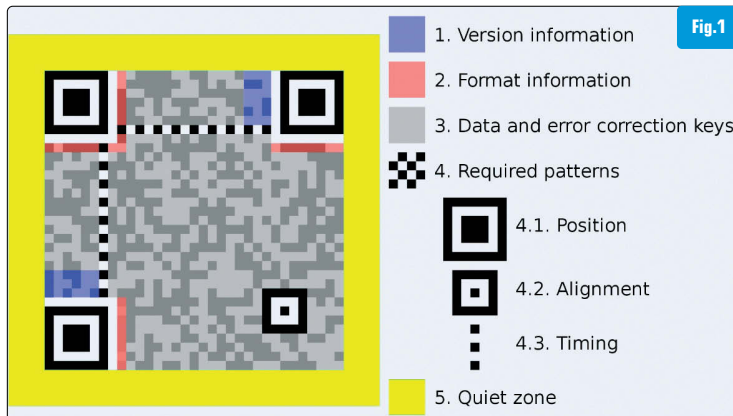
Les QR codes sont constitués de modules. Certains sont fixes : yeux de positionnement (modules entourés d'un carré et placés dans les coins), yeux d'alignement (petits yeux) utilisés pour corriger la perspective induite par le scanner, timing (barres en pointillés joignant les yeux de position), etc. Ces modules fixes permettent au scanner de déterminer l'orientation du QR code et d'en établir le quadrillage.

Les autres zones contiennent le message codé, de la réplication de données ou encore des informations techniques (elles aussi redondées) : numéro de version, format utilisé, encodage, masque, etc.

Enfin, bien que ce ne soit pas absolument indispensable en pratique, les QR codes sont entourés d'une zone vide (blanche) qu'on nomme « quiet zone ». Cet espace sert de délimitation avec le reste du document (sur lequel le code est imprimé) pour faciliter la lecture. On notera simplement que l'efficacité de certaines applications de lecture diminue fortement lorsque la quiet zone est réduite ou absente. La norme prévoit qu'elle soit large de quatre modules [Fig.1].

La taille du QR code, exprimée en nombre de modules, dépend de la nature (ISO, UTF-8, chiffres, etc.) et de la quantité d'informations à coder ; plus le message est long et plus il faudra de modules pour le représenter. Ainsi il faudra beaucoup plus de modules pour représenter une vCard (carte de visite avec de nombreux champs) qu'une simple URL. Les spécifications prévoient quarante versions (tailles) allant de 21 modules de côté à 177.

Dans sa taille la plus petite, un QR code ne peut contenir que 25 caractères alphanumériques. Dans sa plus grande taille, en fonction de l'encodage utilisé, un QR code peut contenir 7089 chiffres, ou



Structure d'un QR code

4296 caractères alphanumériques, ou 2953 octets ou encore 1817 kanjis. Cela nécessite alors 46 yeux d'alignement et 62 modules de timing. A titre de comparaison, un code barre ne permet de stocker qu'une vingtaine de caractères.

Le QR code contenant ma carte de visite (cf. signature de cet article) n'utilise que six yeux d'alignement. Les yeux de position, quant à eux, tous toujours au nombre de trois et disposés dans les coins. Tout comme le Compact Disk (CD), pour lequel Philips a prévu qu'une partie des données du support puisse être détruite (le CD

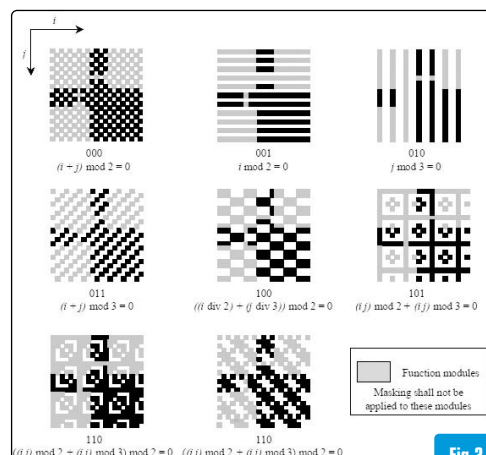
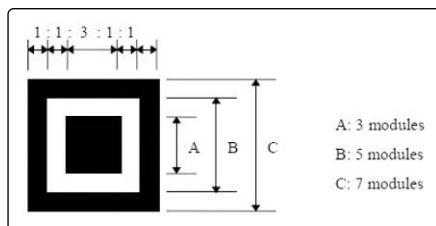
“ On préfère les modules uniformément répartis...”

peut supporter des rayures d'un millimètre d'épaisseur) car elles sont dupliquées ailleurs sur le disque, le QR code intègre une certaine redondance de l'information. Le QR code reste ainsi lisible même s'il est déchiré ou recouvert d'une tache d'encre.

La norme prévoit plusieurs niveaux de « qualité » allant de L (Low-7% de redondance) à H (30%) en passant par M (Medium-15%) et Q (25%). Lorsqu'on emploie le niveau H (High), cela signifie que 30% de la surface utile du QR code correspond à de la redondance d'informations.

Pour un message donné, cela veut aussi dire que plus le niveau de qualité est élevé, et plus le nombre de modules nécessaires est grand. Pour une taille d'impression précise (par exemple 3 cm de côté), un niveau de qualité fort entraîne donc l'utilisation de modules plus petits. Or plus les modules sont petits et plus la caméra (lecteur du smartphone) aura de mal à les scanner. De manière générale, l'étalement des paliers des niveaux de qualité fait que le niveau Q (Quality) est souvent un bon compromis.

L'algorithme de génération des modules peut conduire à des images laides ou illisibles. La principale anomalie est d'avoir un pâté de modules



Masques prédéfinis

(noirs) agglomérés dans lequel on a du mal à compter le nombre exact de modules verticaux et/ou horizontaux. Y en a-t-il quatre ou cinq ? Une autre anomalie courante est de générer des motifs géométriques interdits, comme des yeux. Un œil supplémentaire empêcherait à coup sûr de déterminer l'alignement du code. Enfin, et pas seulement pour des raisons esthétiques, il est préférable que les modules soient uniformément répartis sur la surface.

Pour lutter contre ces anomalies, on superpose un masque (calque), possédant des modules bien placés, par-dessus le QR code et on conserve les modules résultant d'une opération booléenne XOR entre le masque et le code. La norme prédéfinit huit calques (cf. image) répondant à des fonctions mathématiques simples, à base de modules [Fig.2].

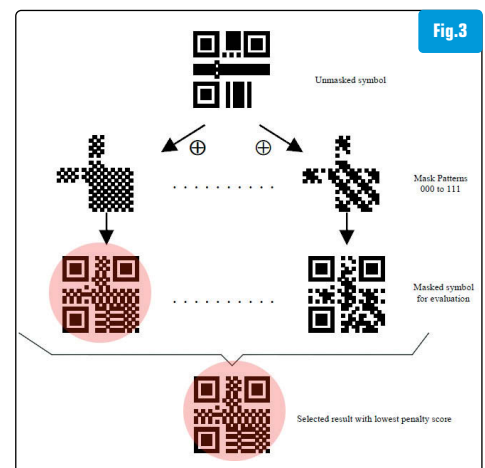
Lors de la génération, le programme doit tous les essayer et conserver celui qui donne le plus beau résultat. Plus exactement, on garde celui qui est le moins laid à l'aide d'algorithmes de pénalités. Le moins laid est celui qui obtient la pénalité la plus faible. L'identifiant du masque choisi (« 000 » dans l'exemple) est bien entendu inscrit dans le QR code [Fig.3].

La bibliothèque la plus populaire pour créer des QR codes (mais aussi des codes-barres, des Data Matrix, des EAN, des ITF, etc.) est sans doute ZXing. Elle permet de générer la matrice des modules en quelques lignes :

```
final String message = «http://www.programmez.com»;  
  
// Creation de la matrice  
final QRCode qrcode = new QRCode();  
Encoder.encode(message, ErrorCorrectionLevel.Q, qrcode);  
final ByteMatrix matrix = qrcode.getMatrix();
```

A partir de cette matrice booléenne, on peut créer l'image finale du QR code à l'aide d'une API graphique comme AWT ou Java 2D :

```
// Modules  
final Area area = new Area();  
final Area module = new Area(new Rectangle.Float(0, 0, 1, 1));  
final AffineTransform transform = new AffineTransform();  
final int width = matrix.getWidth();  
  
for (int i = 0; i < width; i++) {
```



Utilisation des masques

```
for (int j = 0; j < width; j++) {
    if (matrix.get(j, i) == 1) {
        area.add(module);
    }
    transform.setToTranslation(1, 0);
    module.transform(transform);
}
transform.setToTranslation(-width, 1);
module.transform(transform);
}

// Taille
final int size = 400;
final double adjustment = width / (double) (width + 8);
final double ratio = size * adjustment / width;
transform.setToTranslation(4, 4); // Quietzone
area.transform(transform);
transform.setToScale(ratio, ratio);
area.transform(transform);

// Traitement de l'image
final BufferedImage image = new BufferedImage(size, size, TYPE_INT_RGB);
final Graphics2D graphic = (Graphics2D) image.getGraphics();
graphic.setPaint(new Color(0x0000FF)); // Modules bleus
graphic.setBackground(new Color(0xFFFFFFFF)); // Fond blanc
graphic.clearRect(0, 0, size, size);
graphic.fill(area);
```



[Fig.4]
QR code créé par le programme

> Des codes design

De plus en plus, on voit apparaître des QR codes personnalisés, utilisant des modules arrondis, des yeux géométriques (formes diverses), des jeux de couleurs, des dégradés, etc. Les exemples les plus complexes intègrent même des logos ou des graphiques.

“ La France est bien représentée...”

Sur ce point, la France est relativement bien représentée puisqu'Unitag, une société leader dans la génération et la personnalisation des QR codes, est implantée à Toulouse. Le code Java proposé plus haut a d'ailleurs été gracieusement offert par leur équipe de développement. Il est possible de changer la couleur des modules car c'est le contraste avec le fond du QR code qui compte. Ainsi, sur un fond blanc, la plupart des teintes peuvent convenir à condition de ne pas être trop claires.

On peut également insérer un petit logo dans le QR code sous réserve d'utiliser un niveau de qualité élevé (Q ou H). Les modules corrompus par cet ajout sont compensés par la redondance des données. Il convient néanmoins de ne pas modifier les yeux de position et les zones techniques (version, format, masque, etc.) et de tester la lisibilité. Si besoin on pourra déplacer le logo dans le code pour qu'il soit plus lisible et plus esthétique [Fig.5].

Enfin, il est possible de créer des QR codes utilisant des modules blancs sur fond noir (ie. des pixels clairs sur fond sombre). Ils sont

alors appelés « codes inversés ». Toutefois, tous les lecteurs ne savent pas forcément les interpréter.

> Détection et correction d'erreur

Contrairement au protocole TCP/IP (Transmission Control Protocol/Internet Protocol), qui permet de détecter des erreurs de transmission mais qui nécessite l'envoi d'une nouvelle trame pour avoir l'information correcte, le QR code intègre un mécanisme de correction. Lors du scan du QR code, on ne peut évidemment pas demander une réimpression de ce dernier s'il contient une erreur, d'où l'obligation d'avoir un système de corrections. L'algorithme utilisé est le « Reed Solomon » (du nom de ses inventeurs Irving S. Reed et Gustave Solomon) dont le Compact Disk emploie une variante. Son principe repose sur la construction de polynômes à partir de l'information source, que l'on va enrichir. C'est cet ensemble qui est alors envoyé à la place des caractères originaux. Cet ajout surcharge l'image finale du QR code mais en garantit le bon décodage.



Fig.5 QR code design en forme de lapin

“ Simple mais d'une efficacité diabolique...”

Supposons qu'il faille protéger le triplet (a, b, c). Pour cela on va ajouter de l'information sous la forme du couple (S1, S2) à l'aide de fonctions mathématiques bien choisies :

$$S1 = a + b + c$$

$$S2 = a + 2b + 3c$$



Fig.6 Déformation oblique lors du scan



Fig.8 Distorsion de l'objectif

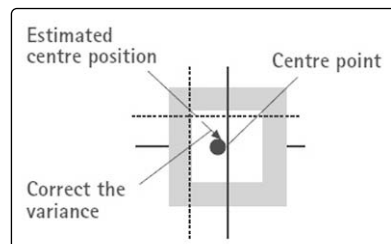


Fig.7 Alignement



Fig.10 QR code design d'Unitag

Pour le triplet (12, 7, 21), on va ainsi ajouter le couple (40, 89) :

$$S1 = 12 + 7 + 21 = 40$$

$$S2 = 12 + 2 \times 7 + 3 \times 21 = 89$$

Imaginons que, pour des raisons diverses, la valeur lue de « b » soit corrompue ; on lit « 10 » au lieu de lire « 7 ». Le couple (V1, V2) résultant devrait donc être (43, 95). La différence ? entre S1 et V1 quantifie l'erreur détectée :

$$? = V1 - S1 = 3$$

Reste à savoir sur quel élément E du triplet porte cette erreur. Ici, l'algorithme suppose qu'un seul élément soit corrompu, ce qui est le plus probable.

$$E = (V2 - S2) \div ? = (95 - 89) \div 3 = 2$$

C'est donc le second élément qui possède un décalage (positif) de « 3 ». Il ne reste plus qu'à corriger les erreurs lues :

$$a = 12$$

$$b = 10 - 3 = 7$$

$$c = 21$$

En pratique, la simplicité apparente du Reed Solomon est d'une efficacité diabolique, ce qui lui a valu une adoption forte par l'industrie. Cet algorithme ne peut néanmoins pas corriger les erreurs multiples (sur plusieurs éléments, se compensant en partie) ou les divergences trop importantes. Les statistiques d'utilisation indiquent que ces cas restent relativement rares.

> Les bonnes pratiques

La plus grosse difficulté que rencontrent les scanners est liée aux déformations de l'image. Personne ne tient son smartphone à la verticale exacte du QR code. La capture de l'image se fait toujours de manière oblique, sur plusieurs axes à la fois. Le QR code apparaît donc en perspective (cf. image). En outre, l'image est généralement

déformée du fait même de la lentille optique du capteur. Cette dernière est conçue pour prendre des photos à une distance donnée. Les clichés pris de plus près ou plus loin font donc apparaître des distorsions, soit vers l'extérieur (cf. image) soit vers l'intérieur [Fig.6, 7 et 8].

Les yeux de positionnement et d'alignement servent à compenser ces déformations de l'image. Il est donc important qu'ils soient lisibles. Cela implique de ne pas en changer la fonction lors de la phase de personnalisation mais surtout que le support d'impression soit le plus plane possible.

La pire situation est quand le code est imprimé sur un support courbe, il devient difficilement lisible.

Ainsi on voit parfois des flyers collés sur les poteaux téléphoniques ou les abris de bus. Ceux-ci étant ronds, le QR code devient quasi illisible. Le logiciel doit alors d'abord utiliser de puissants algorithmes pour redresser l'image [Fig.9].

En outre, les scanners imposent une distance minimale pour réaliser le focus (la mise au point de l'image). Les smartphones de dernières générations disposent tous d'autofocus assez performants mais il ne faut pas oublier les utilisateurs qui ne changent pas de matériel chaque année.

Et c'est sans compter la qualité du capteur utilisé... Cela implique que le QR code soit imprimé dans une taille assez importante et qui dépend directement de la distance du sujet.

Par exemple, un QR code imprimé sur une affiche sera scanné de plus loin que le même code dans un magazine. Il devra donc être imprimé plus gros.

Unitag propose un guide des 10 commandements du QR code sous la forme d'un petit livret en téléchargement libre sur son site. On y découvre notamment l'impact du niveau de qualité sur l'image finale, sa lisibilité sur un support donné, les personnalisations les plus adaptées selon le contexte, etc [Fig.10].

Le livret présente également des exemples concrets de campagnes marketing réussies ou ratées. On comprend ainsi que l'ajout d'un QR code sur des publicités affichées dans le métro et renvoyant vers le site Web du fabricant n'a que très peu d'intérêt dans un espace clos où le Wifi (et donc la connexion Internet) est quasi absent.

A l'opposé, le livret insiste sur l'importance de proposer des contenus de qualité derrière le QR code, faute de quoi son intérêt est limité. La plupart des smartphones disposent d'applications de lecture des QR codes, soit de façon native soit à partir de leurs « app stores » respectifs. En plus de bloquer régulièrement à propos des derniers QR codes qui valent le coup d'œil, le site QR Dress Code présente une sélection des meilleurs générateurs de QR code en ligne, classés par fonctionnalités, et des scanners les plus performants. Un dernier point à garder en tête est le type de scanner utilisé. Ainsi, quand le QR code renvoie vers un site Web, et que le lecteur cible est un iPhone, il faut que le site soit consultable depuis le terminal. Cela implique de ne pas utiliser la techno Adobe Flash (illisible sur iPhone) et d'avoir une présentation adaptée au format (petit) de l'écran.



Carte de visite (vCard) avec des yeux de position en croix

Thierry Leriche-Dessirier
Architecte JEE freelance
<http://www.icauda.com>

Unitag : <http://www.unitag.fr>
QR Dress Code : <http://www.qrdresscode.com/>

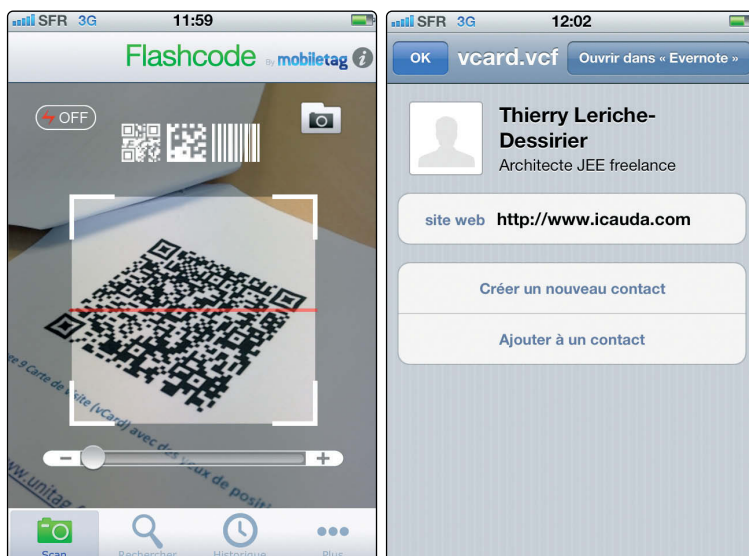


Fig.9 Scan oblique en cours sur iPhone

La conquête des applications **mobiles hybrides** et l'émergence d'un nouveau standard **multi OS**

L'écosystème de la mobilité professionnelle a longtemps gravité autour du quasi monopole de Windows Mobile. Les acteurs spécialisés dans les usages de la traçabilité, la dématérialisation des processus, la gestion des tournées fournissaient le bon terminal, l'infrastructure, les applications et le service pour intégrer et exploiter la nouvelle organisation mobile.

Dès la fin des années 1990, les services et applications mobiles se multiplient, surfent sur le déploiement des réseaux mobiles et sur l'offre croissante de terminaux.

Apple provoque alors une rupture sur ce marché, dès 2007, en introduisant un smartphone qui :

- banalise les technologies et fonctionnalités les plus avancées,
- plante le décor d'un nouveau marché qui permet de proposer rapidement des centaines de milliers d'applications.

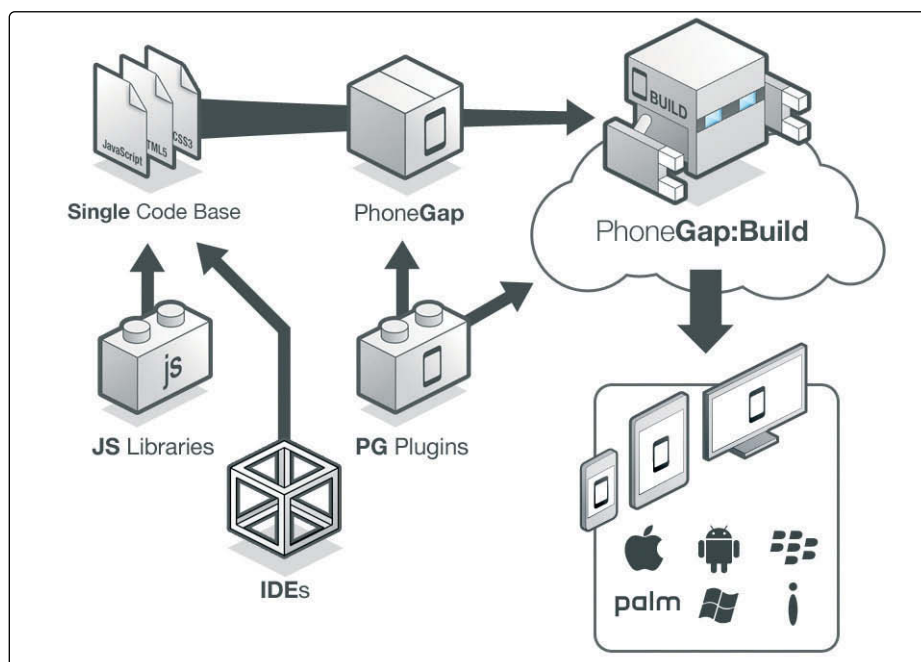
> La rupture

Dans ce contexte, d'autres acteurs économiques majeurs (Google, Nokia, Samsung, RIM, Microsoft...), issus de cultures et business modèles contradictoires s'adaptent pour constituer désormais un ensemble d'écosystèmes non compatibles pour le développement et le déploiement de services mobiles.

Le marché de la mobilité est drivé à la fois par les terminaux du marché, leurs fonctionnalités et les nouveaux écrans, mais aussi par la créativité des éditeurs d'applications qui indépendamment des constructeurs doivent rencontrer au meilleur prix le plus grand nombre d'utilisateurs quel que soit le terminal utilisé.

Dans ce contexte, on observe plusieurs points clés:

- L'efficacité du parcours client sur le mobile est destiné à capter l'attention des clients mobinautes, à emporter leur décision d'achat et soutenir le programme de fidélité. Cette efficacité tient souvent davantage à la simplicité des applications qu'aux technologies embarquées par les mobiles.
- Les stratégies de développement et le choix des outils permettent de rester audible auprès de tous les terminaux.



> Le tiercé gagnant

Aujourd'hui, 80% des applications et services mobiles disponibles sur le marché sont des applications natives focalisées sur l'OS ciblé ou des sites web plus ou moins adaptés à un usage sur le browser d'un terminal mobile.

La combinaison de l'expérience utilisateur avec HTML5 et les frameworks associés couvrent l'essentiel des besoins du marché et anticipent même sur les contraintes de distribution des services mobiles.

Demain, 80% des applications seront hybrides ou des webapps, et toujours focalisées sur l'expérience utilisateur. La tendance clairement observée actuellement est le développement d'applications basées sur les capacités offertes par HTML5 :

- soit sous forme d'applications dites « hybrides », packagées pour être vues comme des applications natives sur le terminal ciblé.

- soit sous forme d'une « webapp », c'est-à-dire une application qui s'exécute dans le browser du terminal, mais avec une expérience utilisateur très proche de celle d'une application native.

Cette tendance forte est basée sur les capacités de plus en plus avancées offertes par le HTML5 et les frameworks associés qui transforment le code HTML5 en un code natif (PhoneGap...)

Cette tendance est confortée par l'engagement d'acteurs majeurs comme IBM, Adobe ou encore Microsoft ; elle se traduit notamment par l'existence d'un nombre de plus en plus important d'applications hybrides sur les différents « stores » applicatifs.

> Les sites mobiles

Dans une proportion encore trop grande de cas, les sites web « mobiles » n'ont pas réellement fait l'objet d'adaptations, ce qui les rend dans la pratique quasiment inutilisables

sur un terminal mobile. La bonne pratique, aujourd'hui, consiste à concevoir les sites afin qu'ils s'adaptent automatiquement au facteur de forme du terminal de consultation, en fournissant une expérience utilisateur réellement efficace.

C'est ce qu'on appelle le « responsive design ». Le parcours utilisateur est facilité par une disposition et un enchaînement des contenus et des interactions adaptés au facteur de forme du terminal utilisé.

> Les "webapps"

Une "webapp" est une application qui s'exécute dans le browser du terminal mobile.

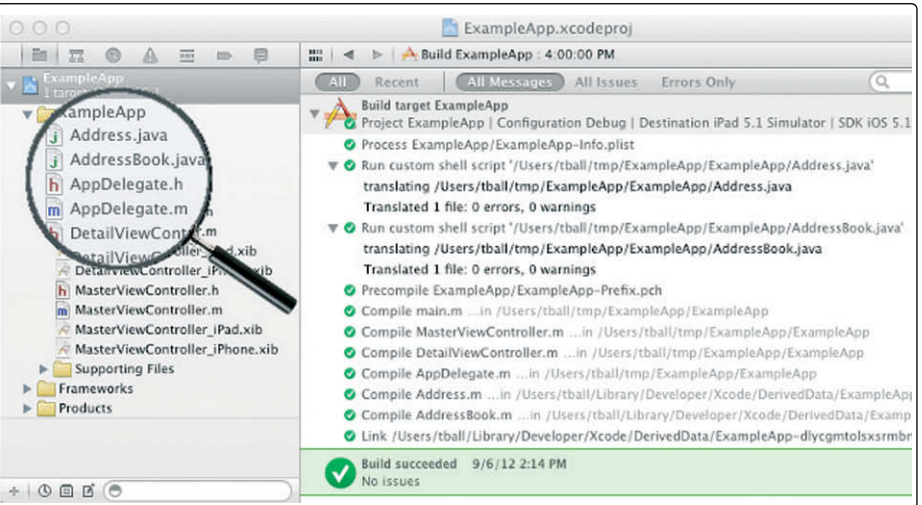
A la différence des sites web mobiles « classiques », et grâce aux capacités offertes par le HTML5, les "webapps" s'appuient sur des interactions utilisateur similaires aux interactions offertes par les applications mobiles natives ou hybrides.

Les principaux intérêts des "webapps" résident dans :

- leur fonctionnement identique sur l'ensemble des terminaux mobiles dont le browser supporte le HTML5
- la simplicité de leur mode de diffusion (une adresse url)
- la suppression des problèmes de mise à jour

Les "webapps" présentent cependant des limitations, qui doivent être considérées en fonction de l'usage ciblé :

- les « webapps » sont plus dépendantes de



la qualité de la connexion réseau, même si des optimisations sont possibles, allant jusqu'à des possibilités de fonctionnement « offline »

- dans des usages très spécifiques, l'expérience utilisateur sera toujours meilleure avec une application native
- en l'état actuel des capacités du HTML5, la géo localisation est possible au sein d'une webapps, mais pas l'accès aux fonctions du téléphone : appareil photo, accéléromètre...

> Les applications hybrides

Les applications hybrides sont des applications mobiles basées sur HTML5.

Elles sont développées à l'aide de frame-

works JavaScript, tels que Sencha Touch, jQuery Mobile, Dojo Mobile... Le code ainsi obtenu est ensuite transformé en code natif (iOS, Android, BlackBerry...) à l'aide de frameworks spécialisés tel que PhoneGap. PhoneGap fournit les bibliothèques de « plugins » permettant d'intégrer dans les applications hybrides l'usage de l'ensemble des fonctionnalités des terminaux mobiles : appareil photo, annuaire, ... L'avantage principal des applications hybrides est de permettre le développement en un seul code d'applications mobiles à destination de terminaux mobiles équipés d'OS différents. Les expériences utilisateurs et les performances obtenues permettent d'envisager un usage relativement large des applications hybrides.

> Les nouveaux standards

Du « must » au « spécifique » le statut des applications natives évolue et dans le même temps les applications dites « hybrides » deviennent un « standard multi OS ».

La première étape est de s'affranchir des contraintes liées aux terminaux et à leur OS. La seconde étape sera de lever les contraintes de publication des appstores publics qui fonctionnent plus comme des « péages » ou même « octrois virtuels » que comme des « têtes de gondole ». L'enjeu est de développer des private stores capables de collecter et de servir des statistiques d'usages à haute valeur ajoutée pour les Responsables Marketing, particulièrement sensibles aux comportements des mobiles.

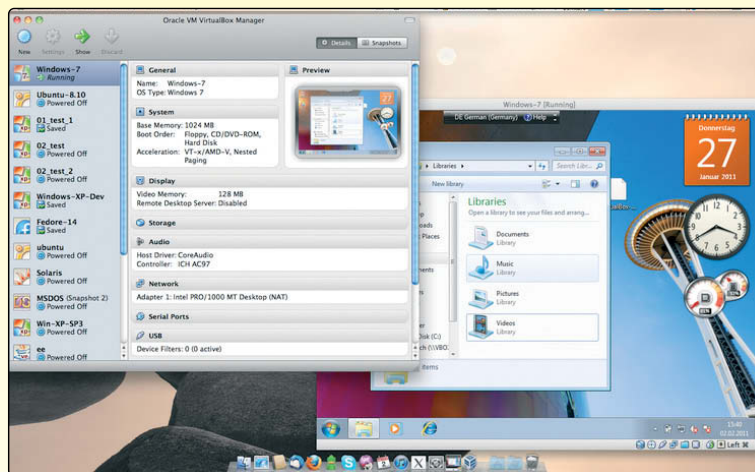


Remy **Buclin**, Corporate Business Developpeur Mobile & Digital.



Virtualisation : l'amie du développeur

“ Non, la virtualisation n'est pas pour moi » ou encore « la virtualisation pour le développeur ? Vous plaisantez ? », combien de fois ai-je entendu ce genre de remarques ! Aujourd'hui, la virtualisation peut rendre de très précieux services aux développeurs que ce soit sur le poste de travail, en déplacement, pour les phases de tests ou tout simplement pour tester de nouveaux outils ou systèmes. La virtualisation sert à beaucoup de choses et facilite la vie du développeur. Toujours pas convaincu ? Nous allons voir quelques scénarios possibles et dresser un panorama des offres, payantes ou gratuites, de virtualisation. Enjoy !



Pourquoi utiliser la virtualisation quand on est développeur ? C'est une question que j'entends régulièrement quand j'évoque la virtualisation au quotidien. En réalité, il existe différents scénarios d'usages particulièrement pertinents, que ce soit sur son poste de travail (ou un poste secondaire) ou sur un portable.

poste de travail	<ul style="list-style-type: none"> - tester un outil, un système en bêta ou pour évaluation - monter des targets de tests, des environnements complets des systèmes cibles - disposer de différentes versions d'un système, d'une application - ne pas multiplier les machines
poste nomade	<ul style="list-style-type: none"> - disposer d'un environnement d'exécution à montrer - ne pas polluer son système avec des builds non stables - idéal pour des démos, des conférences
au niveau des équipes, de l'entreprise	<ul style="list-style-type: none"> - déployer un environnement identique à l'équipe - créer des laboratoires de tests virtuels - ne pas polluer le poste de travail du développeur / testeur - ne pas multiplier les machines

> Donner de la flexibilité à votre machine !

La virtualisation offre un confort incomparable aux développeurs et testeurs. Bien souvent, nous avons besoin de tester un nouveau système, un IDE, une technologie précise. Au lieu de le faire d'une seconde machine, il suffit de créer un master d'image virtuelle et de le dupliquer en fonction des besoins. En un clic (ou presque), vous démarrez et stoppez la machine virtuelle.

Surtout, si vous travaillez sur Windows et que vous avez besoin d'un Linux ou d'un Windows plus ancien, on crée une machine virtuelle et on l'installe dedans. Ou encore mieux, il existe aujourd'hui de très nombreuses machines virtuelles pré-packagées (appliance). Il suffit de la télécharger et de l'exécuter ! Notre article sur un usage concret d'un développeur Windows travaillant sur Mac illustre parfaitement cette souplesse...

Sur les laboratoires de tests virtuels, il existe des solutions packagées pour automatiser cette approche.

> Oui mais...

Comme toujours, tempérez un peu votre enthousiasme. La virtualisation desktop exige un minimum de ressources matérielles : un processeur rapide (et au moins 2 cœurs), 4 Go de ram minimum (si vous utilisez plusieurs VM simultanément, mieux vaut se fixer 12 à 16 Go), un disque dur à forte capacité et avec des performances élevées en tour / minutes ou sinon, idéalement, un disque SSD 128 ou 256 Go. La GPU n'est pas à négliger, ni la qualité et la taille de l'écran. Bref, comme toujours, plus la machine sera performante, mieux ce sera. Si vous travaillez en infrastructure VDI ou avec un hyperviseur sur un serveur, méfiez-vous des performances réseaux. Mieux vaut de l'Ethernet que du wifi par exemple. Le panorama des offres est sur la page suivante.

François Tonic

Une offre large et pour tous les goûts

Quand on parle virtualisation, il faut distinguer plusieurs situations : applications, desktop, serveurs. Ce sont les trois grandes catégories même si nous pouvons aussi rajouter la VDI. Nous ne parlerons pas ici des hyperviseurs.

La plupart des développeurs utiliseront des solutions de virtualisation pour leur poste de travail. Cela signifie que l'outil va permettre de créer, gérer et exécuter des machines virtuelles sur votre machine, ou sur un portable, etc. La VM est portable, donc peu de soucis quand vous l'emportez avec vous (sauf à faire attention aux disques virtuels). Il existe de simples lecteurs de VM.

> Méfiance sur le format

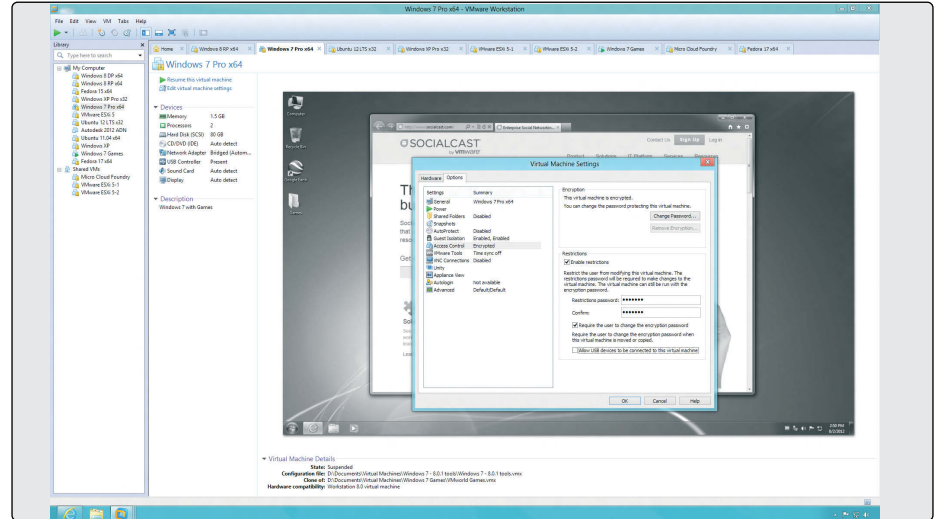
Comme toujours en informatique, en virtualisation, il existe plusieurs formats de VM : VDI (VirtualBox), VMDK (VMware), VHD (Microsoft), HDD (Parallels), QED / QCOW (Qemu). En soi cela ne cause pas de réels soucis. Mais tous les outils ne supportent pas forcément tous les formats de VM du marché. Par exemple, VirtualBox supporte VHD, VMDK, HDD (mais pas les dernières versions). Des outils de conversion peuvent simplifier la portabilité et la conversion. Par exemple, qemu img va permettre de convertir des images VHD, VDI, VMDK. Bien souvent, l'outil de virtualisation va privilégier son format mais permettra un support (complet ou non) de formats tiers (souvent après conversion).

> Une offre variée

Finalement, vous n'avez que l'embarras du choix qu'il s'agisse d'open source / gratuit, ou d'outils commerciaux :

- Open source / gratuit : VirtualBox (Oracle), Qemu, KVM (supporté activement par Red Hat).
- Outils commerciaux : Workstation / Fusion (VMware), Parallels Desktop / Workstation (Parallels).
- Côté Microsoft, jusqu'à Windows 7, on dispose de Virtual PC et du XP Mode (pour les anciennes applications). Windows 8 ne propose plus cette fonction. Le nouveau système intègre un client Hyper-V. Il faut posséder, au minimum, l'édition professionnelle.

Les éditeurs de virtualisation ne proposent pas d'éditions ou de packages spécifiques pour les développeurs. Certains d'entre eux vont par contre fournir quelques conseils ou des choix de solutions. Ainsi, Parallels va conseiller la version Desktop, voire, des solu-



tions serveurs et entreprises pour des usages en équipe ou plus lourds. VMware met en avant l'interface Workstation avec des outils de profiling comme Intel Vtune. Bien entendu, attention à la configuration matérielle. En-dessous de 4 Go de mémoire vive et d'un processeur récent, n'ayez pas beaucoup d'illusions quant aux performances...

> Lequel choisir ?

Le choix va dépendre de nombreux facteurs : le système de travail, les besoins, les fonctions (ex. : support d'outils de profiling dans la VM, snapshot, gestion des ressources, support des formats, etc.). Pour notre part, nous utilisons deux principaux outils : VMware Fusion sur OS X et VirtualBox sur Windows / Linux.

> Nos choix VirtualBox 4.x

La dernière version de VirtualBox (Oracle) supporte Windows 8 et OS X 10.8. La gestion de multiples VM s'améliore (en groupe). La gestion réseau devient plus performante (jusqu'à 36 cartes réseaux, nouvelles fonctions pour les réseaux virtuels). Un gros progrès a été fait sur les fonctions snapshot et de clonages. Et son interface est plutôt agréable à l'usage. D'autre part, il est disponible sur de nombreux systèmes. Sur la page officielle Oracle VM, des appliances sont disponibles. Une version entreprise l'est aussi auprès d'Oracle (39 € / utilisateur, avec sup-

port officiel). Sites : <http://www.oracle.com/tech-network/server-storage/virtualbox/downloads/index.html> Et la version open source : <https://www.virtualbox.org>

VMware Workstation 9

Sans contestation, l'outil phare du marché. Très complet sur les fonctionnalités et les performances, la version 9 supporte Windows 8, USB 3, étend le support matériel, intégration et migration des VM avec vSphere et cloudfoundry, et des améliorations sur la partie 3D. Seul le prix peut dissuader les développeurs : 193,50 € (prix dégressif). Site : <http://www.vmware.com/fr>

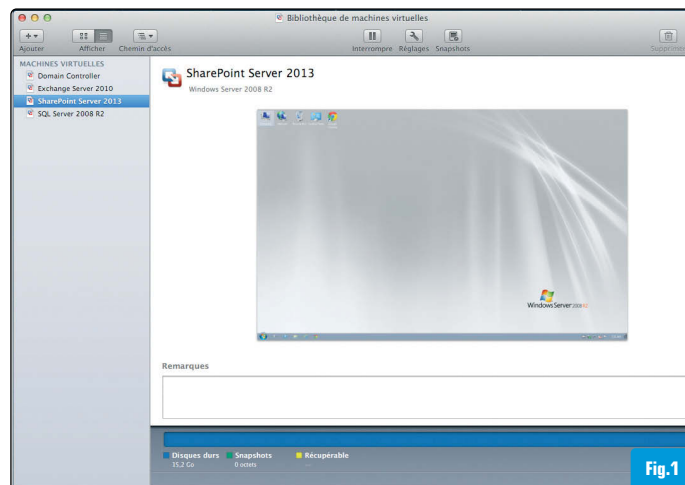
> Le choix du cloud

Les machines virtuelles sur le cloud, un grand classique. Les services IaaS sont nombreux et à des prix intéressants : Windows Azure, Amazon EC2, etc. Il s'agit d'une solution souple et rapide pour déployer des environnements de tests, monter et démonter des scénarios ou tout simplement pour tester un logiciel, un système. Les offres « Développement / Tests » se multiplient rapidement (IBM, HP, Microsoft) pour faciliter les batteries de tests par exemple et surtout pour économiser des ressources machines internes et de l'argent. C'est un choix possible qui fonctionne très bien. Et vous pourrez avoir les mêmes VM sur le cloud que sur votre poste de travail.

François Tonic

Virtualisation sur OS X pour développer sur Windows

Quoi qu'en pensent les plus férus défenseurs d'Apple, Microsoft est encore aujourd'hui majoritaire dans le monde de l'informatique, et il est donc logique que les développeurs travaillant sur OS X se retrouvent tôt ou tard à devoir développer des applications (ex : applications Web) à destination d'utilisateurs Windows sur plateforme .NET.



Depuis de très nombreuses années déjà, la virtualisation est entrée dans le quotidien de bon nombre d'entreprises et notamment de développeurs, afin de leur permettre de réduire les coûts d'acquisition et de possession des matériels qui pour la plupart étaient sous-exploités, ainsi que pour mettre à leur disposition tous les environnements (Windows, Linux, OS X, mobiles...) nécessaires aux tests de leurs applications.

Depuis son adoption des processeurs Intel, et pour permettre aux développeurs Mac de pouvoir basculer vers le monde Windows, Apple a intégré dans OS X le logiciel Bootcamp qui permet de pouvoir passer d'un monde à l'autre, mais avec un inconvénient majeur : devoir redémarrer sa machine à chaque fois.

Généralement, les développeurs n'aiment pas perdre leur temps à redémarrer leur ordinateur et veulent bénéficier du meilleur des deux mondes, notamment quand on leur demande de tester que leur application/site web soit compatible à la fois avec le monde PC (Internet Explorer) et Mac (Safari).

Dans la suite de cet article, nous allons passer en revue les solutions à disposition sur Mac et voir comment utiliser celles-ci afin de développer des applications .NET (ex : application WinForms, site ASP.NET MVC...) sur plateforme Apple.

> Les solutions existantes du marché

La première décision à laquelle sera confronté un développeur sera de choisir un outil de virtualisation, lui permettant de pouvoir rester sous OS X, tout en exécutant Win-

dows (ou autre) au sein d'une machine virtuelle pour lui permettre de réaliser son application.

Dans le monde de la virtualisation sur PC, deux solutions sont généralement proposées à savoir l'utilisation d'un Hyperviseur qui fait office de serveur de virtualisation centralisé (VMware vSphere, Microsoft Hyper-V...) ou bien les logiciels mono utilisateur (VMware Workstation, Microsoft Virtual PC...).

Dans le monde Mac, seule la seconde catégorie était représentée jusqu'à une période très récente. Les solutions entrant dans cette famille sont multiples mais on distingue aujourd'hui trois acteurs principaux :

- VMware Fusion
- Parallels Desktop
- Oracle VM Virtualbox

Les deux premières solutions sont les leaders incontestés du marché et existent depuis de très nombreuses années. Elles viennent toutes les deux d'être mises à jour afin d'assurer la prise en charge de Windows 8 qui arrivera le 26 Octobre prochain. Le troisième acteur est moins connu (toutes proportions gardées), malgré la notoriété de son détenteur, mais il possède un avantage indéniable, à savoir qu'il est totalement gratuit (ses deux concurrents sont vendus entre 45 et 80 €).

En ce qui concerne les Hyperviseurs permettant de créer un serveur hébergeant les machines virtuelles de plusieurs utilisateurs, à ce jour une seule solution est disponible : **Parallels Server**.

Le but ici n'est pas de comparer les avantages et inconvénients de chaque solution ou de mener des débats enflammés comme on

peut le voir dans certains forums de discussion sur le fait qu'un tel soit plus rapide que tel autre ou possède une meilleure assistance technique, mais de voir comment mettre en œuvre une solution de virtualisation sur OS X.

Pour la suite de cet article, nous utiliserons donc VMware Fusion pour mener à bien cette tâche mais la quasi intégralité de ce qui sera évoqué, s'applique également à Parallels ou Virtual Box.

> Fonctionnalités des solutions de virtualisation

Une des particularités des solutions de virtualisation, est de pouvoir exécuter plusieurs environnements de manière simultanée sur une même machine physique, si les ressources matérielles de l'ordinateur hôte (processeur, mémoire, espace de stockage...) le permettent, bien évidemment.

Quand vous exécuterez le logiciel choisi pour la première fois, vous vous retrouverez généralement en face du tableau de bord qui vous permettra d'accéder à l'ensemble des machines virtuelles disponibles sur votre ordinateur, avec la possibilité d'en créer de nouvelles, de démarrer/arrêter celles existantes, de modifier leurs paramètres, etc.

[Fig.1].

La première étape consistera donc à créer une machine virtuelle en définissant les paramètres concernant la configuration matérielle de la machine que vous souhaitez mettre en œuvre. Ces paramètres portent sur tous les critères importants d'un ordinateur :

- Nombre de processeurs
- Quantité mémoire vive
- Nombre et taille des disques durs

- Cartes réseaux et manière de se connecter à l'ordinateur hôte (NAT ou Bridge)
- Support d'une carte son
- Gestion des périphériques USB et des imprimantes

D'autres options avancées sont généralement proposées comme l'intégration des menus de la machine virtuelle dans les menus de OS X, l'accès simplifié aux répertoires de l'ordinateur hôte au sein de la machine virtuelle, le cryptage des données... mais ces options ne sont pas vitales (bien que très utiles) à la mise en œuvre de la solution de virtualisation.

> Gestion des ressources matérielles

Comme évoqué précédemment, la définition des ressources matérielles allouées à la machine virtuelle est importante et c'est pour cela que c'est la première étape dans la configuration d'une nouvelle machine.

Plus vous allouerez de ressources (mémoire notamment) à votre machine virtuelle, plus celle-ci sera réactive, vous permettant ainsi de gagner en efficacité dans votre travail. Il faut toutefois garder à l'esprit que les ressources allouées à votre machine virtuelle, sont autant de ressources en moins dont vous disposerez sur votre ordinateur physique, sachant que celui-ci utilise déjà une partie des ressources pour exécuter OS X et les autres logiciels et services s'exécutant en arrière-plan.

Si vous possédez 16 Go de mémoire et que vous affectez 10 Go à votre machine virtuelle, cela signifie qu'il ne restera que 6 Go disponibles pour OS X et tous les autres logiciels. Par exemple, si vous souhaitez ouvrir Photoshop et faire un lourd traitement sur une photo, vous risquez de constater que ce traitement prendra plus de temps que d'habitude.

Il n'y a pas de règles clairement définies pour savoir quelle quantité de mémoire allouer à telle ou telle machine virtuelle. Tout dépend des logiciels que vous installerez à l'intérieur et des tâches qu'effectueront ces logiciels.

Si nous prenons l'exemple de machine virtuelle nommée **SharePoint 2013**, présente sur la capture d'écran précédente, celle-ci possède 8 Go de mémoire vive, car la future version du logiciel de Microsoft requiert cela pour s'exécuter en plus d'être installée sur un serveur Windows 2008 R2 au minimum.

Si en revanche vous installez une

machine virtuelle avec uniquement Windows 7 et Office 2010 pour effectuer de la bureautique ou utiliser Internet Explorer pour tester la compatibilité d'un site web, alors 2 ou 3 Go de mémoire seront largement suffisants pour ce type d'utilisation.

L'autre point important dans la gestion des ressources concerne l'espace de stockage. Quand vous serez amené à créer un disque dur pour votre machine, vous aurez la possibilité de choisir entre la création d'un disque dynamique ou bien de pré-allouer l'espace.

La création d'un disque dynamique a l'avantage de prendre un minimum de place sur votre disque dur physique, la taille du disque dur virtuel n'augmentant qu'en fonction des données que vous stockerez à l'intérieur (ex : installation de logiciels, copie de fichiers...). A l'inverse, si vous supprimez des données de la machine virtuelle, l'espace en question ne sera pas libéré sur le disque physique tant que vous n'aurez pas compressé le disque virtuel en question (option généralement disponible dans les paramètres de la machine virtuelle).

La pré-allocation de l'espace de stockage a l'avantage d'être un peu plus performante (l'espace étant déjà alloué sur le disque physique) mais avec l'inconvénient d'occuper de l'espace inutile, si vous ne remplissez pas l'espace que vous avez alloué. En clair, si vous effectuez une pré-allocation de 64 Go mais que vous ne stockez que 30 Go de données, les 64 Go seront occupés quoi qu'il arrive.

Comme vous l'aurez compris, mettre en œuvre une solution de virtualisation performante nécessitera donc deux capacités importantes sur votre ordinateur : posséder beaucoup de mémoire vive si vous voulez exécuter plusieurs machines virtuelles de manière simultanée, et beaucoup d'espace de stockage si vous préférez le système de pré-allocation d'espace au système de disque dynamique.

Nous n'avons pas abordé les besoins en termes de processeur car de nos jours, tous les processeurs ou presque (Core i5,

Core i7) sont suffisamment véloces pour encaisser la charge requise par la virtualisation, ceci sans sourciller même avec plusieurs machines virtuelles exécutées en simultanée.

> Installation du système d'exploitation

Une fois la configuration de votre machine virtuelle effectuée (ressources allouées), vient ensuite l'étape d'installation du système d'exploitation comme vous pourriez le faire sur n'importe quel ordinateur de bureau ou serveur que vous achèteriez pour répondre au même besoin.

Différentes solutions s'offrent à vous en fonction du média sur lequel vous possédez le système d'exploitation que vous souhaitez installer. Avec la dématérialisation des supports qui est devenue commune depuis plusieurs années, les fichiers ISO (copie conforme d'un CD mais dans un fichier unique) sont devenus monnaie courante et souvent privilégiés par bon nombre d'utilisateurs. Vous pouvez malgré tout préférer les supports physiques, voire dans certains cas les clés USB bootables sur lesquelles l'équivalent d'une image ISO a été copiée. Quelle que soit la solution que vous choisirez, vous serez à même d'installer le système d'exploitation en vous rendant dans les paramètres de la machine virtuelle et en éditant la section adéquate.

Si vous partez sur un scénario fichier ISO ou CD physique, il vous suffit de vous rendre dans la section CD/DVD et d'éditer les paramètres pour pointer vers la bonne source, comme vous pouvez le voir sur la capture d'écran [Fig.2].

Il est toutefois bon de noter que sur ses derniers modèles (Macbook Pro Retina et Mac Mini), Apple ne fournit plus de lecteur CD afin de privilégier l'aspect dématérialisation et sa boutique d'application en ligne (Mac AppStore). Cela orientera forcément votre choix de manière plus directive puisque vous n'aurez que le choix entre fichier ISO ou clé USB bootable.

Une fois le média contenant le système d'exploitation attaché correctement à votre machine virtuelle, il ne reste plus qu'à démarrer celle-ci et suivre le processus classique d'installation d'un système d'exploitation (qui diffère quelque peu selon Windows XP, Windows 7, Windows 8, Windows Server... donc nous ne nous attarderons pas sur ce sujet).

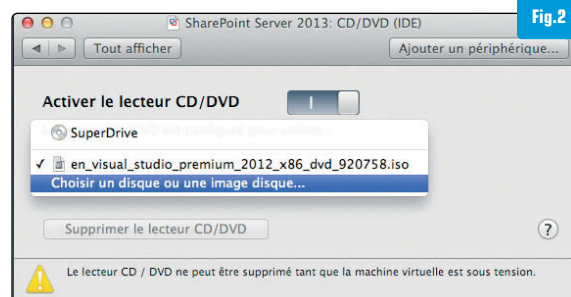


Fig.2

> Installation des logiciels

Une fois l'installation du système d'exploitation terminée, la dernière étape avant de pouvoir commencer à utiliser son nouvel environnement pour développer des applications avec Visual Studio 2012, faire de la bureautique avec Office 2010, etc., il ne reste qu'à installer lesdits logiciels.

Tout comme pour l'installation du système d'exploitation, il n'y a aucune différence dans la démarche, à savoir que si vous possédez le logiciel au format ISO, il suffit d'attacher le fichier dans les paramètres CD/DVD et si vous possédez le fichier au format EXE, il suffit d'exécuter celui-ci comme n'importe quel programme sous Windows en double cliquant dessus.

Il faut garder à l'esprit que bien qu'on parle de machine virtuelle, vous possédez désormais un ordinateur Windows pleinement fonctionnel, avec le même comportement que si vous étiez face à un ordinateur physique.

La seule différence est que vous êtes dans une fenêtre qui s'exécute sous OS X et que vous pouvez passer d'un environnement à l'autre en 1/10^e de seconde. Si vous le souhaitez, vous pouvez même totalement faire croire que vous êtes sous Windows en passant en mode plein écran (Ctrl+Cmd+Enter sur VMware Fusion).

Comme vous pouvez le voir, nous avons une machine virtuelle qui s'exécute sur Windows 2008 R2, sur laquelle Visual Studio 2012 a été installé et qui est actuellement en cours d'exécution [Fig.3].

> Développement et test d'un site ASP.NET MVC

Comme nous l'avons évoqué précédemment, une des volontés (même si ce n'est pas la seule) qui poussera les développeurs à vouloir virtualiser un environnement Windows sous OS X, concernera le fait de vouloir vérifier la compatibilité d'une application web avec le monde Apple et plus particulièrement Safari pour Mac (qui possède quelques différences par rapport à la version PC).

Prenons donc ce scénario et regardons comment faire en sorte de pouvoir tester un site hébergé au sein de la machine depuis OS X et le cas échéant, effectuer un débogage en règle (même si en théorie un développeur ne fait jamais de bogues dans son application 😊).

Nous ne nous attarderons pas sur le développement du site web à proprement parler

puisque ce n'est pas le but de l'article. Nous partirons donc du modèle ASP.NET MVC 4 fourni en standard dans Visual Studio 2012 et vérifierons ensuite que ce site s'exécute déjà correctement au sein de la machine virtuelle (en l'exécutant sous Internet Explorer). Notre site fonctionnant, voyons comment rendre celui-ci accessible à OS X. Pour cela, il faut s'attarder sur les paramètres de la carte réseau de la machine virtuelle (accessible dans les paramètres comme pour le CD/DVD).

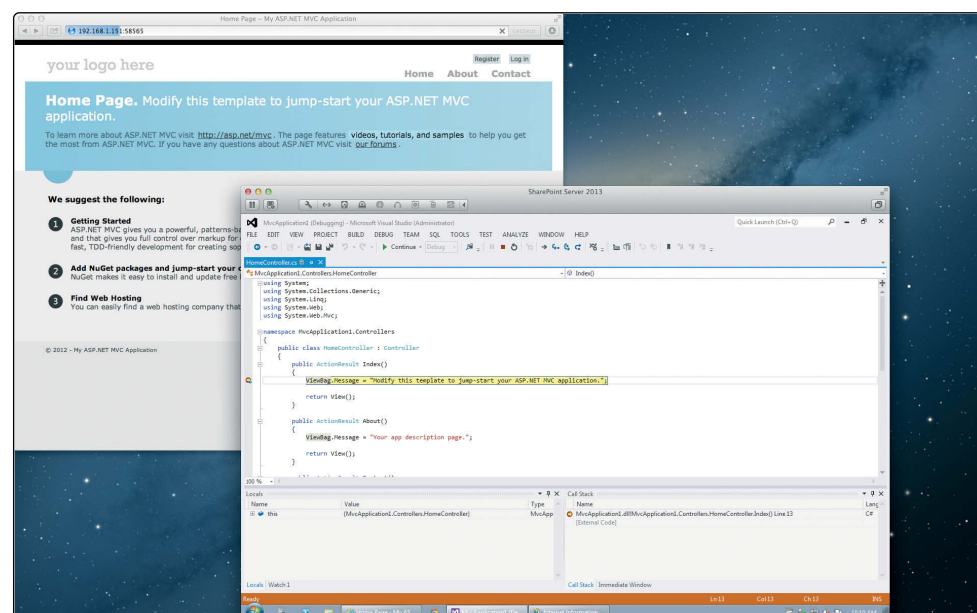
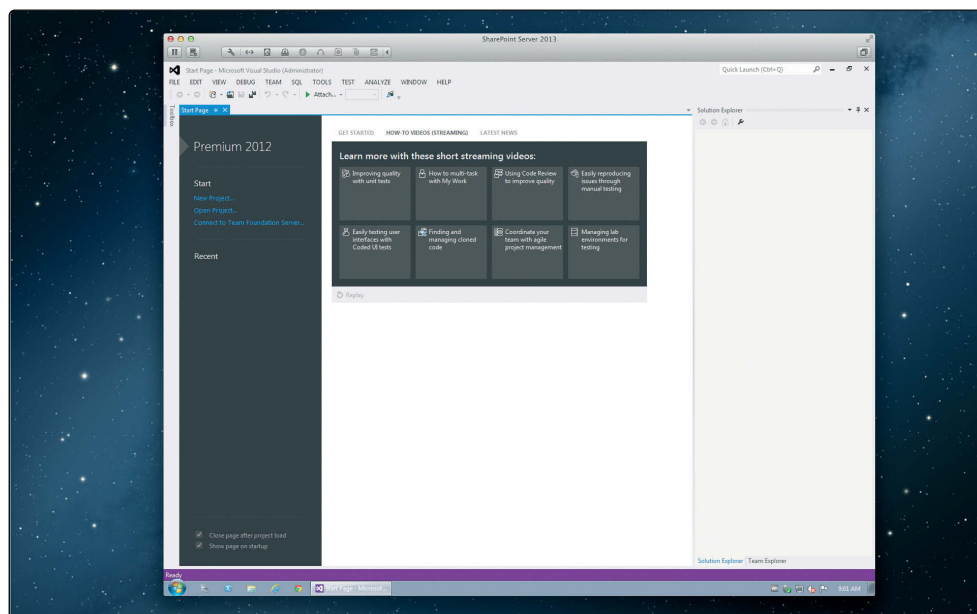
Trois options sont habituellement disponibles :

- Hôte uniquement
- NAT
- Pont (Bridge en Anglais)

La première option permet de créer un réseau virtuel entre toutes les machines virtuelles qui peuvent s'exécuter simultanément

sur l'ordinateur physique. Cette option est très utile quand vous voulez recréer une infrastructure complète (ex : Active Directory, SQL Server, Exchange Server...) mais sans interférer avec le reste du réseau. Tout ce qui est connecté au réseau virtuel, ne peut communiquer qu'avec le réseau virtuel. La seconde option permet à l'ordinateur physique, de servir de passerelle entre la machine virtuelle et le reste du réseau (en utilisant les fonctions de translation d'adresse ou NAT en Anglais). Vous configurez dans ce cas votre machine virtuelle pour qu'elle utilise un serveur DHCP et c'est le logiciel de virtualisation qui attribue les adresses aux machines et se charge de les faire communiquer avec le reste du réseau local.

La dernière option permet à votre machine virtuelle d'utiliser directement la carte



réseau de l'ordinateur physique. Votre machine virtuelle est considérée dès lors comme faisant totalement partie du réseau local et doit être configurée en conséquence (adresse IP, masque de sous-réseau, passerelle...). Si un serveur DHCP est présent sur le réseau local, votre machine virtuelle peut récupérer une adresse IP directement auprès de celui-ci.

Dans notre cas, nous avons configuré notre machine virtuelle en mode Bridge et avons mis les paramètres réseau adéquats. Notre machine virtuelle possède l'adresse IP 192.168.1.151 tandis que notre ordinateur physique possède l'adresse 192.168.1.12 (d'autres machines sont présentes sur le réseau).

Retournons maintenant dans Visual Studio pour faire une modification sur le projet ASP.NET MVC créé précédemment. Dans les paramètres de celui-ci, nous demandons de ne pas utiliser le serveur web interne mais d'utiliser le serveur IIS qui a été installé au préalable. Nous affectons l'adresse <http://192.168.1.151:58565> au site web (le port a été choisi au hasard car d'autres sites s'exécutent sur le serveur) et par souci de commodité nous utilisons l'adresse IP mais nous aurions tout aussi bien pu utiliser un nom de domaine.

Après avoir vérifié que tout fonctionne tou-

jours correctement au sein de la machine virtuelle, passons côté OS X avec Safari et tentons d'accéder au site via l'URL indiquée précédemment. Si tout se passe correctement, vous devriez voir apparaître le site s'exécutant dans la machine virtuelle, directement au sein de Safari.

Si vous souhaitez mettre des points d'arrêt au sein de Visual Studio pour trouver d'où peuvent venir les problèmes au sein de votre code, vous pouvez le faire sans souci comme vous pouvez le constater sur la capture d'écran où nous voyons en parallèle Safari en train de tenter de charger la page, tout en étant stoppé dans Visual Studio par le point d'arrêt positionné dans le contrôleur de la vue Home de notre site web [Fig.4].

Vous pouvez le voir sur la capture, en fonction de la résolution de votre écran (1920x1080 dans notre cas), il peut être difficile de jongler entre les fenêtres, de les superposer, etc. Quand on travaille sur plusieurs tâches en même temps (Safari d'un côté pour la consultation et la machine virtuelle de l'autre pour le débogage). Il est souvent constaté que les développeurs ont deux écrans pour pallier ces inconvénients.

> Conclusion

Comme nous l'avons vu dans cet article, les solutions de virtualisation sur OS X sont mul-

tiples, afin de permettre aux développeurs du monde Apple, de pouvoir également travailler dans le monde Microsoft pour répondre aux besoins imposés par leur métier et tirer ainsi le meilleur des deux mondes.

Ces solutions sont bon marché (moins d'une centaine d'Euros), rapides à mettre en œuvre et simples à prendre en main, tout en proposant l'ensemble des fonctionnalités nécessaires à la réalisation de toutes les applications imaginables (client lourd, site web, application mobile, plug-in bureau-tique...).

La seule contrainte sera d'avoir une machine assez puissante, surtout côté mémoire vive, pour être à même de pouvoir exécuter plusieurs environnements en parallèle, dans des projets de plus en plus complexes et mettant en œuvre de plus en plus de composants/logiciels différents.

L'émergence des solutions Cloud telles que Windows Azure peuvent également apporter des réponses puisqu'il est désormais possible d'héberger ses machines virtuelles dans les nuages.

Stéphane Cordonnier

Directeur - iTouch

<http://www.itouchconsulting.com>

Formez et fidélisez vos équipes techniques

en leur offrant un

Abonnement à Programmez !

Une formation et une veille technologique, à prix réduit



an d a onnement _ _ seulement par a onné (partir de a onnés)

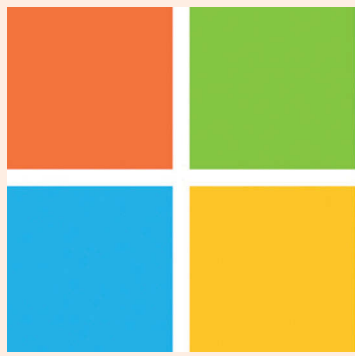
Pour 3,25 € par mois par développeur,

+Offre de lancement : 1 page de **Publicité offerte** pour votre entreprise !

partir de abonnés, o re de lancement

Adresse un mail à diff@programmez.com, en indiquant le nombre d abonnés souhaité, pour recevoir le devis

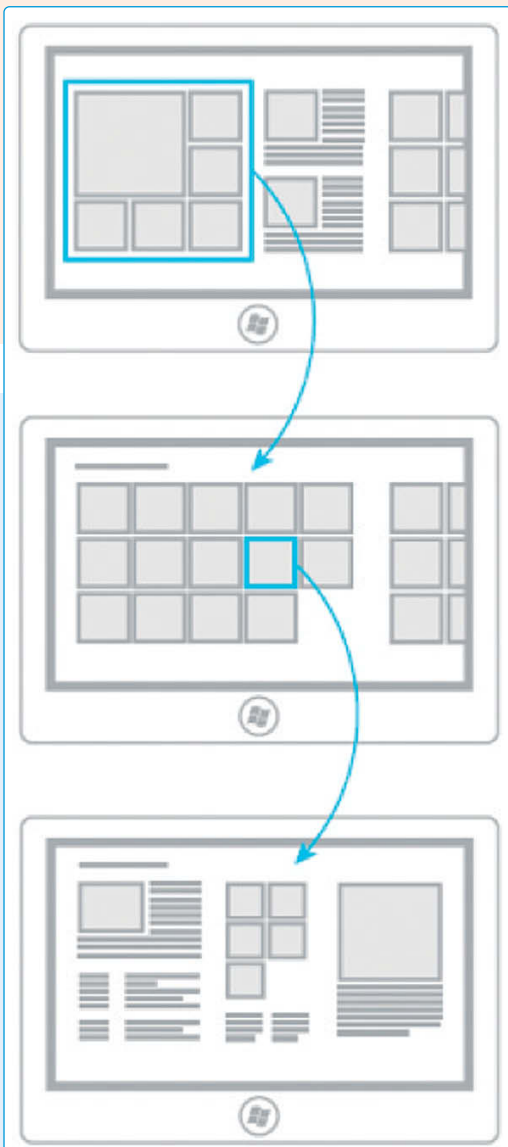
www.programmez.com



Windows 8

de A à W

2^e partie



Le mois dernier nous avons pu aborder plusieurs thèmes très importants pour le développeur : les principales nouveautés du système (API, outils), comprendre Windows Store et commencer à utiliser Modern UI en contexte tactile. Dans cette 2^e partie, alors que Windows 8 est disponible depuis quelques jours, nous allons continuer notre exploration technique (migrer une application de Windows Phone 7 à Windows 8, comprendre les problèmes de chargement des applications, Visual Studio 2012...) et aussi montrer les premières réactions de développeurs qui utilisent et développent sur Windows 8 depuis plusieurs mois. Le défi est immense aussi bien pour Microsoft, et l'ensemble des partenaires, que pour les développeurs qui doivent modifier, adapter les applications. Une phase d'apprentissage est nécessaire pour comprendre WinRT, les API et surtout assimiler la nouvelle interface Modern UI. Sur ce point, Microsoft se veut strict. Et d'ailleurs pour pouvoir passer la validation Windows Store, il faut impérativement respecter les fameuses guidelines officielles.

Voici deux liens absolument incontournables :

- Index of UX guidelines for Windows Store apps (avec un gros PDF de 300 pages) :

<http://msdn.microsoft.com/en-US/library/windows/apps/hh465424>

- Windows 8 product guide for developers :

<http://msdn.microsoft.com/en-us/windows/apps/hh852650.aspx>

A vous de jouer !

François Tonic



Windows 8



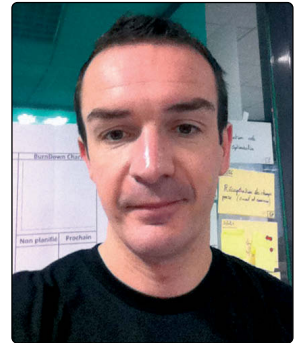
Microsoft

Ce qu'en pensent Les DÉVELOPPEURS...

Témoignages

Entre Visual Studio et Scrum

Arrivé en 2008 chez Chapitre.com au poste de Directeur Technique, j'ai rencontré une personne issue du monde Java qui m'a fait découvrir Scrum. Voyant le grand succès de cette méthodologie, je me suis attelé à faire migrer les équipes .Net sur cette méthode de développement en m'aidant de l'outillage Microsoft.



Un succès qui m'a conduit ensuite chez MediaPost (<http://www.mediapost.fr>), une filiale de la Poste, pour conduire le passage des équipes sous Agile et Team Foundation Server 2010 en m'appuyant sur les compétences et l'accompagnement de Cellenza (<http://www.cellenza.com>), cabinet de Conseil Agile exclusivement dédié aux technologies Microsoft.

Je pratique aujourd'hui en tant que freelance le passage d'équipes en Agile et leur management.

> Outillage Scrum dans Visual Studio 2010

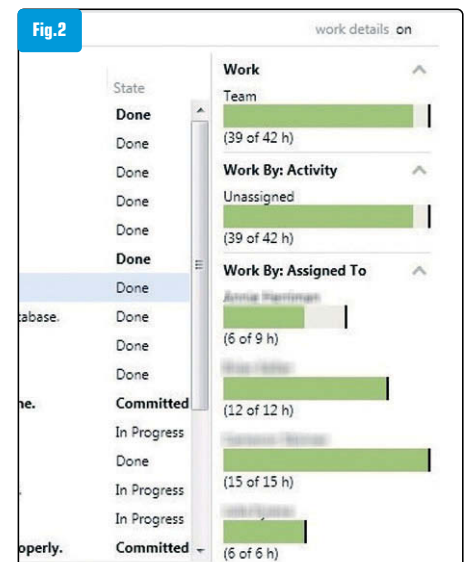
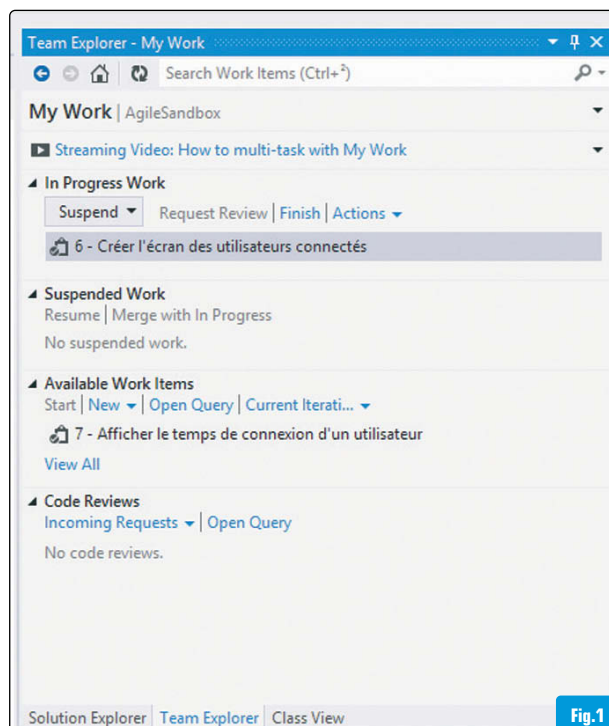
Le site Chapitre.com était développé en ASP.NET avec un backoffice maison en C#, l'ensemble géré sous Team Foundation Server 2010 avec Visual Studio 2010. Le template Agile proposé dans Visual Studio 2010 (« MSF For Agile Software Development 5.0 ») nous paraissait trop éloigné de la méthodologie telle qu'elle est prescrite pour être exploitable. Par conséquent nous avons opté pour le papier et les post-it. Plus tard chez MediaPost, le template « Scrum 1.0 » avait été édité et permettait une exploitation bien plus proche de la méthodologie avec Visual Studio nativement. Grâce à cette intégration de la méthode dans nos

outils de développement, nous avons pu pousser Scrum plus loin dans l'entreprise : utilisation de Sharepoint pour les échanges documentaires, documentation dans le wiki, TFS 2010 pour le contrôle de code source et la gestion de nos backlogs, le Sharepoint TFS pour visualiser nos backlogs. Malheureusement, les outils n'étaient pas encore suffisamment matures et souffraient de défauts de jeunesse : séparation de la gestion documentaire et des backlogs, fonctionnalités présentes dans Visual Studio qui ne sont pas sous Sharepoint, utilisation trop complexe de Visual Studio Team Explorer pour des non techniciens, etc. Cela a

conduit nos interlocuteurs non techniques à ne plus utiliser Visual Studio ou TFS avec certaines conséquences : multiplication des backlogs, mauvaise visibilité sur le travail de l'équipe, manque de communication, difficulté à échanger des tickets (incidents ou autre), tâches sur papier et plus dans TFS, incapacité à sortir simplement des indicateurs pertinents dont les burndown chart.

> Pourquoi Visual Studio 2012 ?

L'arrivée de Visual Studio 2012 a été une excellente surprise. Nous attendions bien sûr des progrès autour de l'outil, mais nous avons été surpris du fait que Microsoft propose désormais un template nativement Scrum. Il s'agit de « MSF For Agile Software Development 6.0 » dont le fondement du template est fidèle à la méthodologie. Les



sites Sharepoint sont unifiés: un seul site web permet de manipuler simplement nos user stories avec la capacité à changer les itérations, les tests d'acceptation. Ce fut la première remarque de notre

Product Owner : enfin il pouvait travailler sur les mêmes outils que nous !

L'intégration des éléments TFS dans l'interface graphique est un autre fort progrès : les développeurs ont accès à leur Scrum Board directement dans Visual Studio et sans changer de fenêtre [Fig.1].

Tous les indicateurs sont visibles directement depuis l'interface graphique, il est devenu beaucoup plus simple de voir et/ou de créer des tâches. Notre Scrum Master a pu voir immédiatement des indicateurs sur son sprint en cours [Fig.2] : répartition du travail entre les équipiers, vélocité, Burndown [Fig.3], etc. qui n'avaient plus besoin d'être créés manuellement!

Ces différentes avancées et l'engouement de nos clients (qui ne sont pas que des développeurs) nous ont permis de supprimer les feuilles Excel, backlog multiples, papiers, etc. qui venaient polluer notre quotidien. Nos équipes peuvent désormais se concentrer sur leur véritable métier : développer et produire des logiciels de qualité.

> Outils de tests

La qualité est un problème que nous avons rencontré de nombreuses fois : l'utilisation de tests unitaires automatisés n'a jamais été bien comprise par nos équipes de développement du fait de la nécessité d'écrire les tests, les compiler, revenir sur le code et les aller-retour incessants que cela provoque. La compilation et l'exécution en temps réel des tests unitaires [Fig.4] permettent désormais aux utilisateurs de voir immédiatement l'impact de leur développement, ce qui est beaucoup plus motivant dans des équipes qui sont en développement dirigé par les tests (Test Driven Development).

De la même manière, nous avons voulu automatiser des tests d'interface sur les sites web que nous développons. Mais quand les



manipulations deviennent trop complexes, les outils ne suivent plus. Nous avons tenté des approches via des outils tiers (Selenium, Telerik, etc.) mais rien ne nous a vraiment convaincu du fait de la quantité de paramétrage qui était nécessaire.

Visual Studio 2012 permet d'enregistrer des scénarios de tests, de les récupérer sous forme de Code C# et de les modifier ensuite permettant ainsi très simplement pour nos équipes la réalisation de tests d'interface entièrement automatisés. C'est pour nous, très certainement, la plus forte avancée qui va enfin nous permettre de passer en TDD.

La possibilité dans Visual Studio de demander facilement, via l'interface du développeur, une revue de code à une autre personne va également nous permettre d'intégrer cette pratique dans nos équipes.

Le déploiement de Visual Studio et Team Foundation Server 2012 que nous envisageons dans les semaines à venir va nous permettre d'enfin unifier notre pratique Agile et nos outils de développement tout en donnant de la visibilité à nos clients et à nos donneurs d'ordre. Nous espérons beaucoup améliorer notre qualité et pouvoir progresser dans notre pratique du TDD via ces outils !

> Des ressources de Qualité pour un Delivery de Qualité

Dans un contexte économique difficile, le pilotage par la valeur, la suppression des gaspillages, le respect des délais et des budgets sont devenus pour Mediapost, comme pour de nombreux donneurs d'ordre, des priorités pour les Directions Etudes.

Le choix de la méthode Scrum et l'accompagnement de Cellenza m'ont donc permis de mettre en place ces principes et d'organiser leur déclinaison pratique au sein des

équipes, en m'appuyant notamment sur un Scrum Master et des Team Members expérimentés.

A coût réduit, ou tout au moins maîtrisé, nous avons également dû produire

autant, tout en préservant la qualité du service rendu aux Directions métier. Dans cette optique, Cellenza a mis en place les outils et les principes nécessaires à l'augmentation de la productivité des équipes de développement.

En outre, la définition d'une architecture optimisée, le calcul et l'amélioration de la vélocité des équipes de développement et la simplification dès que possible ont été également quelques-uns des ingrédients que les consultants de Cellenza ont utilisé pour nous aider à ce que projet soit un succès.

Enfin, la non-qualité a un coût, communément reconnu par les DSI. Dans ce contexte, l'accompagnement de Cellenza a consisté à mettre en place une culture, des pratiques d'ingénierie et des outils de mesure de la qualité au niveau projet destinés à réduire et éradiquer la dette technique, mère de tous les vices et des dérapages budgétaires.

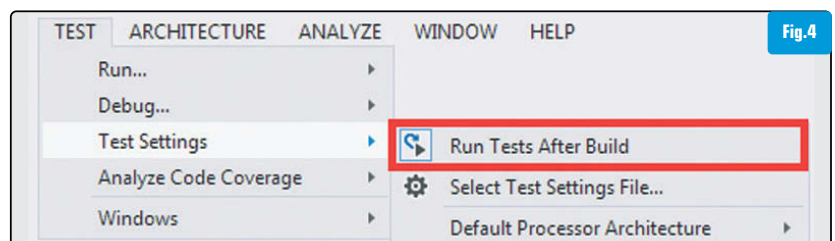
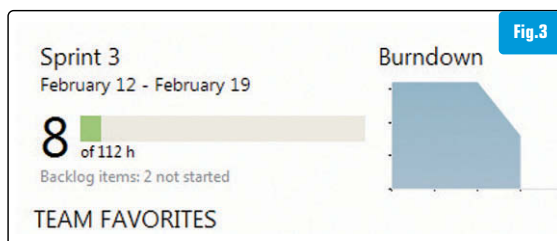


En synthèse donc, la qualité des équipes et leur expertise à la fois technique et méthodologique est à mes yeux tout aussi importante que l'évolution des produits Visual Studio (2010 vs 2012) dans l'atteinte des objectifs et la réussite des projets agiles que je suis en train de mener.

Olivier Congq



Responsable Etudes & Développements chez MediaPost
olivier@agilit.pro



Témoignages

Nathalie Belval

(Direction Technique - Responsable développements internes & Experte XAML, So@t) :
une passion pour Modern UI

Étant passionnée des interfaces interactives depuis 2007 avec WPF puis Silverlight et enfin Windows Phone, j'ai donc ainsi voulu tester Windows 8. Je n'ai pas été déçue au premier lancement qui nous accueille avec son nouvel écran ModernUI et ses tuiles. C'est beau, fluide et personnalisable selon un panel de couleurs assez vaste. On est un peu perdu au début mais après quelques jours on s'y habitue très bien. Après plusieurs mois d'utilisation, on ne peut plus s'en passer !

Les menus cachés aux quatre coins de l'écran permettent de gagner en efficacité. Une fois les nouveaux raccourcis clavier mémorisés, c'est un plaisir. On retrouve toute la puissance de Windows 7 avec une surcouche, nous permettant de passer à une utilisation tactile avec des applications dont les utilisateurs sont de plus en plus friands.

Beaucoup de changement sur Windows 8 ! Mais finalement c'est pour notre bien et notre efficacité,

les applications sont plus centrées sur les besoins de l'utilisateur, tout est connecté (Facebook, Twitter, Mail, etc.) dès que la tablette et /ou le PC est bien configuré, c'est un plaisir de l'utiliser. Les applications de base répondent déjà aux besoins primaires de l'utilisateur (mails, internet, réseaux sociaux et photos), le reste n'est que du supplément !

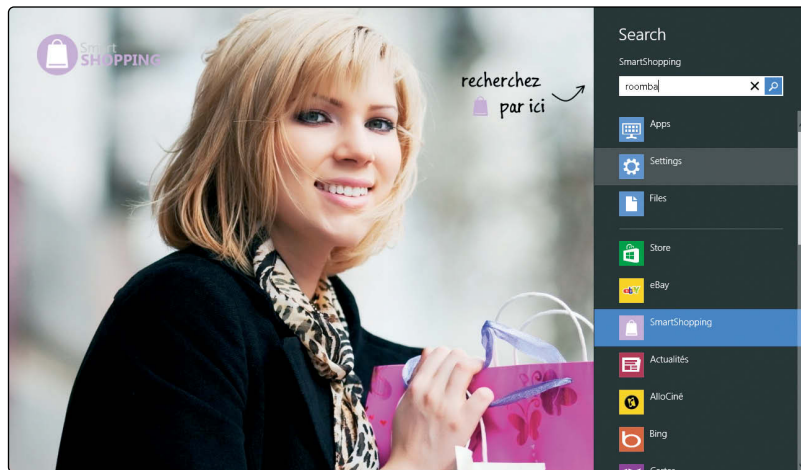
> Les outils de développement

Dans Visual Studio, beaucoup de changements : ceux qui sont habitués à utiliser Expression Blend reconnaîtront les différents panneaux (ToolBox, Document Outline, Properties, etc.).

Un nouveau designer comme sur Blend, permet de modifier les propriétés d'un contrôle de manière aussi détaillée que dans Blend avec les styles, les templates et l'arborescence des contrôles. Maintenant que certaines fonctionnalités de Blend sont

intégrées à Visual Studio, que devient Expression Blend ? Celui-ci conserve tout ce que possède Visual Studio, il reste très utile si vous avez besoin de faire des animations : StoryBoard et VisualState.

De plus, Blend est doté d'une nouvelle fonctionnalité spécifique à WinRT et très pratique. Celle-ci vous permet d'avoir un aperçu de votre application dans les différentes vues pour les Windows Store Apps : FullScreenLandscape, Filled, FullScreenPortrait



et Snapped. L'essentiel des nouveautés pour Blend se trouve plus côté développement HTML avec l'aperçu en temps réel intégré directement dans Blend.

> Retour d'expérience sur le développement

Afin de m'appropriier ces nouveaux outils et l'API WinRT, je me suis donc lancée dans le développement de mon application appelée SmartShopping, publiée depuis quelques jours sur le store.

Cette application permet à l'utilisateur de rechercher un produit qu'il désire acheter afin de trouver immédiatement le meilleur prix en comparant les différents vendeurs. Une prochaine version permettra également de suivre l'évolution des prix en temps réel en utilisant le système de notifications avec Azure.

Lors du développement de mon application, j'ai ainsi pu mettre à profit les fonctionnalités spécifiques à Windows 8 à savoir :

• La recherche :

Pas facile à mettre en place puisque c'est la fonctionnalité principale de mon application et que les guidelines Modern UI contraignent les développeurs à ne pas mettre de bouton de recherche directement dans l'application mais d'utiliser la recherche de Windows 8. Au départ, je ressentais ce point comme une contrainte mais finalement, une fois que les utilisateurs ont pris l'habitude on devient assez vite accro à cette barre de recherche

qu'on utilise de plus en plus au quotidien.

Afin de ne pas perdre l'utilisateur au premier lancement de mon application, qui paraîtrait vide, j'ai donc créé une petite animation qui incite l'utilisateur à utiliser cette barre de recherche et donc de comprendre tout de suite le principe.

• La snapView :

La fameuse vue où on peut lancer deux applications en parallèle. J'ai ainsi réussi à

lui donner tout son sens dans mon application en permettant à l'utilisateur de naviguer sur les différents sites vendeurs afin d'y comparer le produit, frais de port et conditions d'expédition ou paiement.

> Le partage

Là aussi, très pratique pour que les utilisateurs s'échangent des liens vers des produits, cette fonctionnalité était donc essentielle à mettre en place.

> CONCLUSION

Le gros avantage de Windows 8 comparé aux autres OS, c'est qu'il propose des fonctionnalités aux développeurs qui sont généralisées et ainsi très faciles à mettre en place (la recherche, le partage, les autres contrats). Toutes ces fonctionnalités sont réellement un gain de temps et un plus qui permettent vraiment de mettre l'utilisateur au centre de l'application tout comme le fait déjà Windows Phone.

Témoignages

Nathanael Marchand (expert .Net, So@t) est wahou !

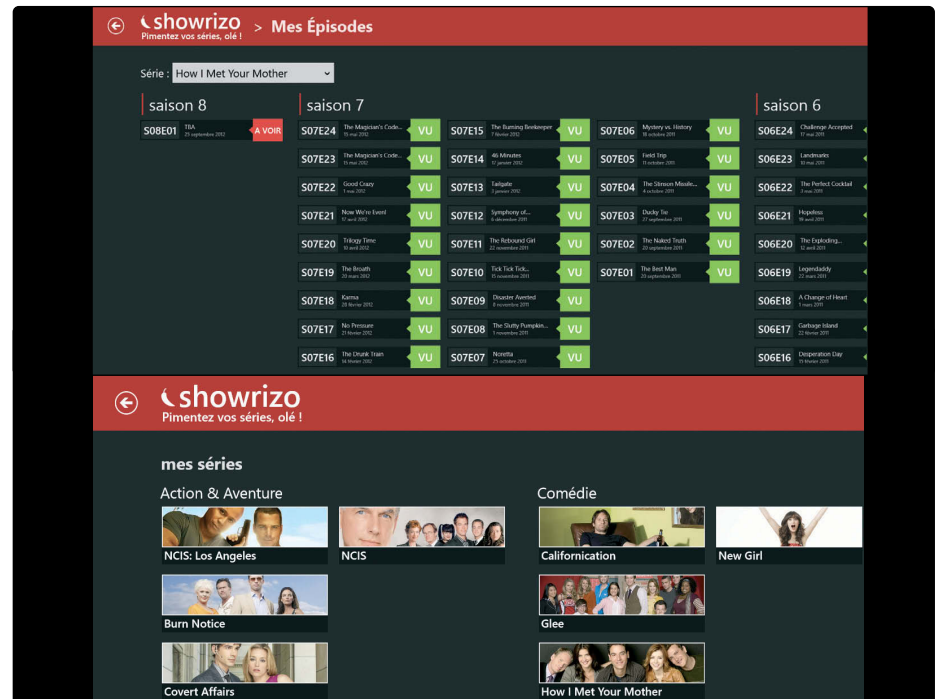
Au premier lancement, wahou ! C'est coloré, c'est acidulé, j'adore. Bon, mon jugement est légèrement biaisé car je suis déjà fan des préceptes Modern UI et je possède déjà un Windows Phone.

J'ai installé le système sur plusieurs machines : mon gros PC fixe, mon ordinateur portable et ma tablette Samsung Slate. Bien que les utilisations ne soient pas les mêmes, ça marche plutôt bien. Après un petit temps d'adaptation pour prendre mes repères et pour arrêter de chercher les différences, l'utilisation commence à être agréable.

> Windows 8, les outils de développement

Je suis un habitué des outils Microsoft, je travaille sur Visual Studio avec .Net et notamment WPF et Silverlight. Pour moi, il était donc tout naturel de choisir C# et XAML. Pour les premiers pas avec les outils, un peu de gêne : le XAML est celui qui était disponible avec Silverlight 3. Il manque beaucoup de choses par rapport à Silverlight 5. Passé cette petite gymnastique, c'est aux nouvelles API qu'il faut s'attaquer. En effet, avec .NET 4.5, il y a pas mal de nouveautés concernant la gestion de l'asynchronisme, par exemple les mots-clefs `async/await`, les nouvelles méthodes d'accès aux fichiers, etc. Première constatation : c'est plutôt bien conçu, un gros coup de balai sur les méthodes obsolètes a été donné avec toujours dans l'optique une gestion asynchrone. Du côté de Blend, pas trop de nouveautés à se mettre sous la dent. L'essentiel est du côté de la gestion du HTML5.

Au niveau des concepts de développements, on est très surpris ! En effet, pour soumettre une application sur le Windows Store, il faut suivre les recommandations Modern UI et le moins qu'on puisse dire, c'est qu'elles sont conséquentes : un PDF de plus de 300 pages ! Dans un premier temps, en plus du volume, on est vite découragé par les contraintes. C'est d'ailleurs un sentiment que j'ai pu constater auprès de



différents développeurs. Finalement, le plus nouveau sur Windows 8, ce ne sont ni les langages, ni les outils mais bien ces recommandations !

> Windows 8, la prise en main des outils

Afin d'appréhender correctement les outils, je me suis mis en tête de développer réellement une application pour voir ce que la technologie avait dans le ventre. J'ai travaillé avec Cyril Cathala, collègue expert .net, sur Showrizo, une application de gestion de planning pour les séries télé. Ça a été également l'occasion de mettre à l'épreuve les outils collaboratifs de Microsoft, à savoir Team Foundation Server (via TFS Preview). Le moins que l'on puisse dire, c'est que le résultat a été concluant. Une fois passés tous les a priori de la nouveauté, on retrouve vite les marques habituelles. On commence à apercevoir et comprendre les différents choix qui ont été faits par les équipes de Microsoft et l'ensemble est plutôt cohérent. Et c'est en utilisant la plateforme qu'on comprend comment développer correctement 😊. Un exemple très simple : les contrats. Cette fonctionnalité permet de définir les interactions entre une application et l'écosystème (autres applications ou machine). Ainsi, pour l'utilisateur, quelle que soit l'application dans laquelle il est, la manipulation pour arriver aux para-

mètres est la même : barre des charmes, l'engrenage des paramètres. Du côté du développeur, cela permet aussi de se poser moins de questions sur « comment je dois présenter mes paramètres ? » ou ce genre de questions. En fait, après un peu d'entraînement et quelques discussions avec les équipes « Division Plateformes et Ecosystèmes » de chez Microsoft, on comprend vite que les consignes Modern UI sont faites pour être comprises mais surtout pour qu'on se les approprie. Cela permet de donner une cohérence et une fluidité à l'utilisateur. Et même, le design n'est pas oublié, il existe des applications très réussies sur le Store, ce sont des équipes qui ont fait la démarche d'intégrer correctement ces consignes.

> Conclusion

Globalement, mon expérience avec Windows est pour le moment très positive. Un système qui reprend les qualités de Windows 7, en lui ajoutant des nouveaux paradigmes (notamment les tuiles, fonctionnalité dont je suis un grand fan). Côté développeur, notre application Showrizo (disponible sur le Store), a été développée dans un temps équivalent à ce qu'il nous aurait fallu pour la développer sur WPF ou Silverlight. Les outils ont été améliorés et le seul petit bémol à y apporter concerne le XAML de Windows 8, légèrement en retard sur ce qu'on peut retrouver en WPF ou Silverlight, ce qui reste un détail...

Prenez une longueur d'avance : créez une application Windows 8 avec Visual Studio & TFS 2012

Partie 6 (fin) : Les Tests

Depuis les versions 2010 de Visual Studio et Team Foundation Server, les équipes disposent d'un outil entièrement dédié à la gestion et l'exécution de tests fonctionnels : Microsoft Test Manager. Cet outil se retrouve donc tout naturellement dans la version 2012 avec, bien sûr, son lot de nouveautés.

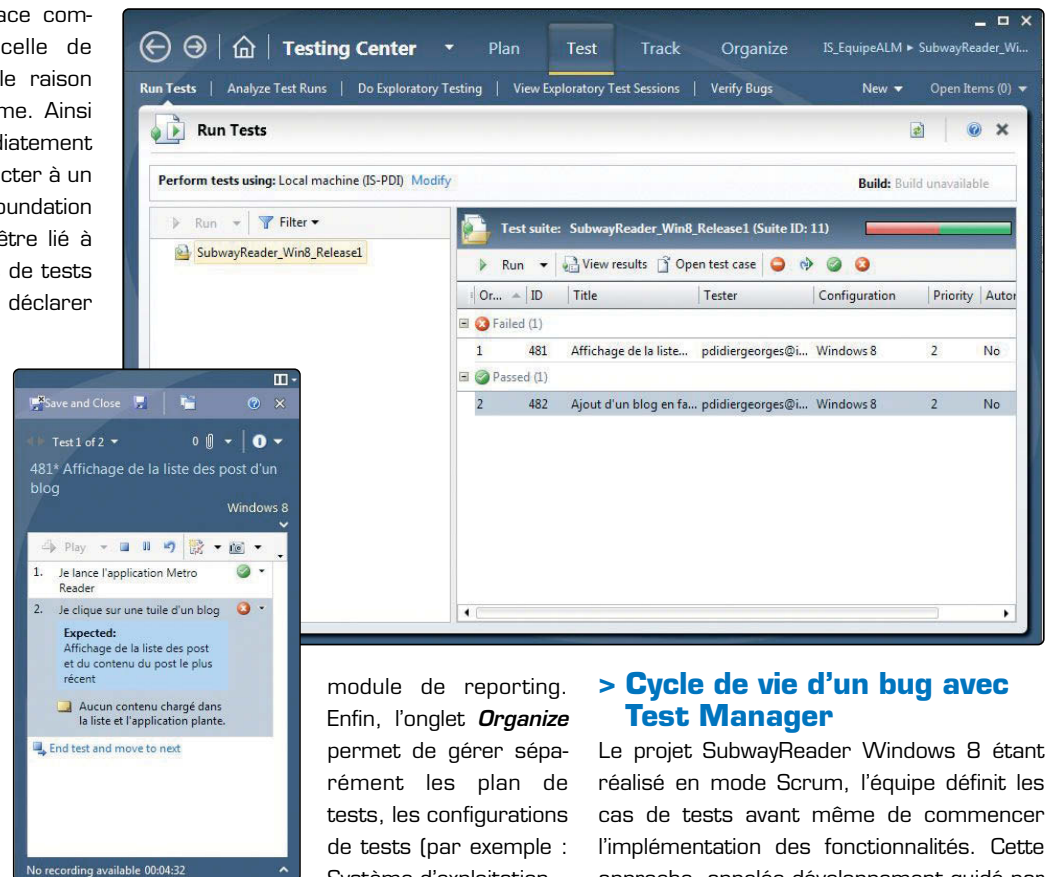
Cet outil propose une interface complètement différente de celle de Visual Studio pour la simple raison que le public visé n'est pas le même. Ainsi au lancement de l'outil, il est immédiatement demandé à l'utilisateur de se connecter à un projet d'équipe sur un Team Foundation Server. En effet il est impératif d'être lié à TFS afin d'accéder aux campagnes de tests existantes, de pouvoir tester ou déclarer des bugs, consulter les résultats des précédents tests, etc.

Une fois connecté au projet d'équipe on accède à la liste des plans de tests. Un plan va permettre de gérer une campagne de tests en ciblant une zone (Area) du projet d'équipe ainsi qu'une itération. Dans le cas de l'application SubwayReader, il est envisageable de créer un plan de test pour la validation de la première Release de l'application Windows 8.

C'est alors ce plan qui sera utilisé pour tester les fonctionnalités à implémenter pour cette release tout au long des sprints la composant.

Une fois le plan de test sélectionné, on découvre l'interface qui se divise en quatre parties. L'onglet **Plan** permet de gérer les plans de tests, et notamment les cas de tests inclus dans la campagne, mais propose également des outils d'analyse des résultats obtenus précédemment.

L'onglet **Test** regroupe les outils permettant l'exécution de tous les types de tests disponibles : cas de tests, tests exploratoires, tests semi-automatisés et vérifications de bugs. L'onglet **Track** va permettre aux testeurs de coordonner les campagnes de tests et les différentes builds en cours. Cet onglet offre en outre la possibilité de voir les tests restant à exécuter et d'accéder au



module de reporting. Enfin, l'onglet **Organize** permet de gérer séparément les plans de tests, les configurations de tests (par exemple : Système d'exploitation = Windows 8), les cas de

tests qui peuvent chacun être inclus dans plusieurs plans et les shared step qui sont des actions communes à plusieurs cas de tests (lancement de l'application par exemple). En définitive, le Test Manager permet de piloter les tests et permet à tout moment d'identifier à quel stade en est une campagne, les taux de réussite et d'échec et par conséquent les points critiques à faire remonter à l'équipe de développement. C'est également avec cet outil que les testeurs pourront, lors de l'identification d'un nouveau bug, le remonter à l'équipe de développement par la création d'un nouvel élément de travail de type bug dans Team Foundation Server.

> Cycle de vie d'un bug avec Test Manager

Le projet SubwayReader Windows 8 étant réalisé en mode Scrum, l'équipe définit les cas de tests avant même de commencer l'implémentation des fonctionnalités. Cette approche, appelée développement guidé par les tests (ou test-driven development), présente plusieurs avantages : le premier étant qu'il permet de déterminer avec précision le comportement souhaité de l'application dans un cas d'utilisation. Le second avantage est qu'au moment où les testeurs doivent tester une fonctionnalité, ils disposent déjà du scénario détaillé de test et des critères de réussite ou d'échec du test. Ainsi pour valider un cas de test, il suffit de le sélectionner dans l'onglet **Test** du Testing Center et de démarrer la session en cliquant sur **Run**. L'outil d'exécution des tests manuels s'affiche alors sous la forme d'un panneau latéral comparable à la vue Snapped proposée par Windows 8 et propose de démarrer le test. Une fois prêt, le testeur clique sur

Start Test et les étapes du scénario sont directement accessibles. Il suffit alors de suivre les étapes définies dans le cas de test et d'en valider le résultat attendu à chaque étape. C'est dans ces sessions de tests que sont identifiés les bugs. En cas d'échec à un cas de test, la session passe à l'état failed et il est alors possible de créer un nouveau bug. Le work item de bug alors créé est automatiquement remonté à l'équipe de développement et son évolution pourra être suivie dans Team Foundation Server.

A sa création, le bug peut également être associé à un cas de test destiné à tester qu'il est bien résolu dans les prochaines campagnes de tests. C'est ainsi qu'une fois le bug traité par un développeur, le cas de test doit être rejoué afin d'en valider le résultat et ainsi passer l'état du bug à *closed*.

> Les tests exploratoires

Les tests exploratoires sont une nouveauté de la version 2012 du Microsoft Test Manager. Il s'agit d'effectuer des tests sans toutefois disposer d'un scénario prédéfini, d'où la notion d'exploration.

Le but de cet outil étant de faciliter la création de cas de tests qui n'auraient pas été prévus avant l'implémentation. Ainsi lors d'une séance de test exploratoires, l'outil utilisé est le même que lors de l'exécution de sessions de tests, à la différence que les étapes sont enregistrées au fil de l'exploration. Le testeur renseigne les étapes et leurs critères de réussite au fur et à mesure qu'il exécute les actions. Au cours de l'exploration, il suffit de cliquer sur *Create new test case* pour générer un cas de test avec toutes les actions enregistrées jusque-là.

Le cas de test créé peut alors être inclus dans un plan de test afin d'être validé comme c'est le cas pour n'importe quel autre. Pour effectuer une session de test exploratoire, il suffit de cliquer sur *Do Exploratory Testing* dans le bandeau de l'onglet *Test*.

AUTOMATISATION DES TESTS MANUELS

> Agilité et automatisation des tests

Lors d'un développement suivant une méthodologie Agile, les tests sont destinés à être joués encore plus souvent. L'utilisation de cycles courts (itérations ou Sprint) demande à valider non seulement les tests concernant les nouvelles fonctionnalités ajoutées, mais également à vérifier que les fonction-

nalités précédentes soient toujours opérationnelles. C'est cette dernière catégorie de tests, souvent appelés **Tests de non régression**, qui gagne fortement à être automatisée. Si l'automatisation des tests doit avant tout être une opération simple, il faut également bien garder à l'esprit que tous les tests de non régression n'ont pas forcément vocation à être automatisés. La logique de vérification de certains tests est parfois beaucoup trop complexe pour être automatisée, là où une simple vérification humaine suffit. À l'inverse, certains tests nécessitent des opérations très simples, mais souffrent parfois de temps de latences conséquents (temps de chargement de plusieurs secondes/minutes entre chaque action, calculs longs ou recherches en bases de données, etc.). La répétition de ces tests à chaque itération est souvent non seulement source de frustration de la part des testeurs, mais mobilise également des ressources pour des opérations à faible valeur ajoutée. En les automatisant, ces tests pourront être rejoués sans mobiliser de testeurs, qui pourront ainsi se concentrer sur l'exécution d'autres tests à plus forte valeur ajoutée.

L'automatisation de ces tests manuels présente également de nombreux avantages pour l'équipe de développement. Elle peut ainsi exécuter à la demande (tous les soirs par exemple) la batterie de tests d'interface automatisés, et ainsi vérifier rapidement, indépendamment de l'équipe de test, que l'application n'a pas subi de régressions.

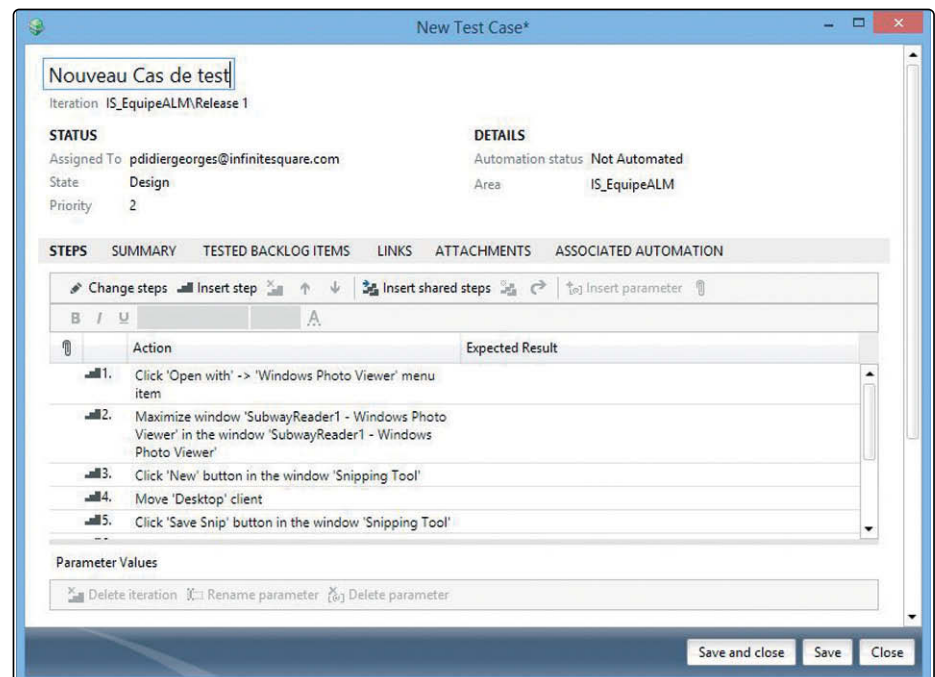
> La solution Microsoft : les Coded UI Tests

La solution d'automatisation de tests manuels de Microsoft s'appelle les **Coded UI Tests** (ou tests codés d'interface utilisateur). La création de Coded UI Tests peut se faire suivant deux scénarios principaux :

- Il est possible de récupérer un enregistrement d'actions effectuées dans Microsoft Test Manager
- Il est possible de démarrer d'un projet de test vide, puis d'enregistrer les actions à reproduire à l'aide du Coded UI Test Builder

La fonctionnalité du **Fast Forwarding** de Microsoft Test Manager s'appuie sur la technologie d'automatisation des Coded UI Tests. Il est alors possible de récupérer l'enregistrement pris lors de l'exécution du test manuel dans Microsoft Test Manager, puis de l'adapter pour en faire un test automatisé, rejouable dans différents environnements.

Pour pouvoir automatiser des tests, il est nécessaire de disposer d'une méthode d'enregistrement simple et puissante, capable d'automatiser rapidement les scénarios les plus simples, sans pour autant se limiter au niveau des possibilités techniques utilisables par le test. Assez logiquement, les Coded UI Tests de Visual Studio 2012 s'imposent comme un outil de choix pour assurer ce rôle. Outre la simplicité d'utilisation de l'outil, son intégration avec les autres produits Microsoft permet d'obtenir rapidement un environnement capable de non seulement rejouer des tests automatisés, mais égale-



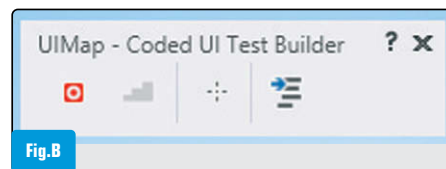
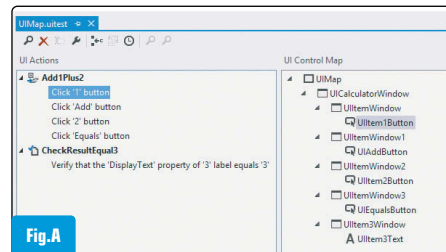
ment de bénéficier des mêmes fonctionnalités disponibles pour les tests manuels :

- Capture d'écran et enregistrement vidéo lorsqu'un bug est rencontré
- Création d'un Work Item de Bug rempli automatiquement des enregistrements du test (Environnement, actions effectuées, Intellitrace)

> Automatisation d'un test d'interface

La création d'un Coded UI Test se fait dans Visual Studio. Dans la version 2012 de Visual Studio, le Template Coded UI Test est disponible, sinon en version 2010, les Coded UI Tests se rajoutent à partir d'un projet de test classique. C'est lors de l'ajout du Coded UI Test dans le projet de test que le choix est proposé entre reprendre un enregistrement effectué dans Microsoft Test Manager, ou partir d'un test vierge. [Fig.A]. L'élément principal de ce test d'interface est sans aucun doute ce que l'on appelle l'UIMap. Généralement représenté par un fichier nommé UIMap.uitest, ce dernier est modifiable via l'outil graphique Coded UI Test Editor (Disponible de base dans Visual Studio 2012, ou en installant le Feature Pack 2 dans Visual Studio 2010). L'UIMap est séparé en deux parties :

- La première (volet de droite) contient l'arborescence des composants visuels qui sont sollicités durant le test Boutons, Images, Texte, etc.)
- La seconde (volet de gauche) décrit les actions qui sont effectuées sur ces composants (Clic gauche ou droit de la souris, appui d'une touche du clavier, etc.)



S'il est possible de modifier ce fichier UIMap dans l'éditeur graphique, il existe une autre manière encore plus simple de l'alimenter : Le Coded UI Test Builder. Disponible par un simple clic droit sur le fichier UIMap.uitest, cet outil possède principalement deux utilisations : L'enregistrement et l'assertion [Fig.B]. Le démarrage de l'enregistrement a pour effet de récupérer toutes les actions

effectuées par l'utilisateur (manipulation souris/clavier), mais surtout identifie de manière hiérarchique les contrôles qui sont employés. Il est important que l'identification se base sur l'arborescence des composants visuels et non pas sur des coordonnées absolues de l'écran, afin d'accepter une certaine tolérance vis-à-vis des modifications de l'interface graphique (ex : Le décalage de 100 pixels sur la droite d'un bouton n'affectera pas un test reposant sur l'arborescence des composants). La création d'assertion est également une étape importante pour la bonne validité du test d'interface. Elle permet de contrôler la valeur d'un champ de texte, ou la présence d'une fenêtre pour valider le bon déroulement du test.

> Pour aller plus loin

La création de Coded UI Tests n'est qu'une étape dans la constitution d'une batterie de tests efficaces. Il est possible par exemple d'employer ces tests dans l'outil de tests de charge de Visual Studio pour pouvoir exécuter des tests de charge à partir de tests d'interface graphique ! Dans ce cas, l'environnement de virtualisation centré sur TFS, le **Lab Manager**, est souvent un outil incontournable de ce genre de scénario. Si la complexité est grandement camouflée par le Coded UI Test Builder et le Coded UI Test Editor, l'outil d'automatisation repose sur les technologies .NET et est très facilement extensible pour peu que l'on possède quelques connaissances en C# ou en VB par exemple.

Philippe Didiergeorges et
Vivien Fabing
Consultants équipe ALM
<http://blogs.developpeur.org/Philess>
<http://blogs.developpeur.org/vivien>

Infinite Square :
<http://www.infinitesquare.com>



FEEDBACK DU CLIENT (Cycle de vie d'une Feedback Request)

Grande nouveauté de cette version 2012 de Visual Studio : L'apparition de la gestion de demande de Feedback appuyée sur un petit exécutable appelé Microsoft Feedback Client. Avec ce nouvel outil, il est maintenant possible d'initier directement une discussion ciblée et rapide entre l'équipe de développement et les clients. La demande de Feedback se fait à partir du Team Web Access et nécessite uniquement trois parties essentielles : Les adresses emails des destinataires, où se procurer l'application (URL Web, partage réseau) et sur quel(s) point(s) focaliser son attention. La première fois que l'on reçoit une demande de feedback, dans l'email est précisée une URL pour se procurer le petit client de Feedback. Une fois installé, on dispose alors d'une interface similaire au Microsoft Test Manager pour pouvoir faire ses commentaires : Capture d'écran, enregistrement vidéo, enregistrement de la voix, saisie de commentaires, etc. tous les outils sont disponibles pour recueillir l'avis de l'utilisateur [Fig.C]. Une fois le Feedback réalisé, toutes les données sont ensuite automatiquement remontées et centralisées sur TFS et peuvent être alors traitées par l'équipe de développement. C'est ainsi une bonne occasion de réajuster certains développements, voire d'alimenter le backlog de nouvelles fonctionnalités détectées durant cette demande de Feedback.

Optimisation du lancement d'une application Windows 8 en XAML/C#

Avec la multiplication des appareils tactiles et connectés au Cloud, les utilisateurs s'attendent désormais à profiter d'applications rapides et fluides qui les informent de ce qu'elles sont en train de faire tout en étant réactives à leurs sollicitations.

C'est afin de répondre à cette attente que nous, développeurs, devons accorder une attention toute particulière aux diverses problématiques liées aux chargements dans l'application. Dans cet article on considérera deux grands types de chargements, le chargement de l'application elle-même et le chargement de ses données.

> Vitesse effective et vitesse perçue

En tant que développeur, on estime généralement qu'une opération est rapide lorsqu'elle met peu de temps à s'exécuter. Considérons une méthode A et une méthode B chargeant une liste de 500 produits pour une application de magasin en ligne. A met 5 secondes pour charger ces données qui sont ensuite affichées d'un bloc alors que B met 15 secondes pour effectuer la même chose. La grande différence entre A et B est que B renvoie les produits téléchargés par palier de 20 produits, on a alors un chargement progressif qui nous permet de peupler la liste des produits au fur et à mesure. L'utilisateur préférera généralement la seconde méthode car, bien que plus lente, il a très tôt de l'information qui s'affiche et a l'impression que l'application est plus rapide. C'est la différence entre la vitesse *effective* et la vitesse *perçue*. Dans le processus de développement de l'application on va donc devoir concentrer nos efforts sur le ressenti de l'utilisateur qui, à la différence d'un temps d'exécution mesuré, est beaucoup plus subjectif.

> Lancement de l'application

Le lancement de l'application est une étape critique pour l'expérience utilisateur. C'est le moment de la mettre dans de bonnes dispositions en s'assurant qu'elle se charge vite. On considère généralement qu'une application qui se lance en moins d'une seconde est rapide, en moins de 5 seconde, bonne mais peut mieux faire et en plus de 5 secondes, mauvaise. Il faut retenir que si elle est trop longue à se charger (plus de 5 secondes) Windows 8 va tuer son processus et de toute façon il y a relativement peu de chances que vous passiez la certification du Windows Store dans un tel cas.

Même si votre application met 4 secondes à se charger, rien n'empêche un utilisateur de considérer cela comme excessif et donc de désinstaller votre application, ce que vous ne souhaitez probablement pas.

NB : Afin de vous assurer que votre application se charge rapidement il est conseillé de tester cela sur un appareil de faible puissance. Alternativement, vous pouvez dans la plupart des ordinateurs portables, configurer la fréquence processeur maximum et donc la brider à 20% ou 30% pour vos tests.

L'amélioration du ressenti de l'utilisateur lors du lancement de l'application peut se faire sur plusieurs axes principaux :

- Optimisation du XAML (Vitesse effective)
- Extension du Splashscreen (Vitesse perçue)
- Optimisation du chargement des données (Vitesse effective et perçue)

> Optimisation du XAML

Le parsing du XAML est une étape assez lourde du démarrage de l'application. Plus la quantité de XAML à parser au lancement est grande, plus l'application mettra longtemps à se charger. Il est indispensable d'optimiser cette partie car c'est une chose que vous ne pouvez pas vraiment contrôler.

Le parsing du XAML utilise essentiellement le processeur pour effectuer son travail. Dès lors, sur un appareil de faible puissance type tablette ARM, on peut aisément et rapidement dépasser les 5 secondes de temps de démarrage réglementaires. Ce temps-là étant fonction de la puissance des appareils, on n'a pas le choix, il va falloir optimiser.

Charger le minimum

Une manière simple et efficace de procéder ici est de ne charger au démarrage de l'application que ce qui lui est strictement nécessaire à ce moment-là. Il va donc falloir faire le tri dans vos ressources.

On a souvent pris la mauvaise habitude de créer un gros fichier de ressources types `ApplicationStyles.xaml` qui contient toutes les ressources nécessaires (et souvent même plus que nécessaires) au fonctionnement de l'application et que l'on référençait directement de la manière suivante :

```
<Application.Resources>
  <ResourceDictionary>
    <ResourceDictionary.MergedDictionaries>
      <ResourceDictionary Source="ApplicationStyles.xaml"/>
    </ResourceDictionary.MergedDictionaries>
  </ResourceDictionary>
</Application.Resources>
```

Ceci est vrai pour tous les fichiers chargés par la première page de l'application.

Chaque dictionnaire de ressources référencé doit être entièrement analysé et toutes les ressources qu'il contient seront instanciées. C'est un processus assez long qui faut limiter au maximum.

Factoriser les ressources

Bien que valable pour toute la durée de vie de l'application, une bonne factorisation de vos ressources peut avoir un effet positif non négligeable sur le temps de lancement. Supposons que la page principale de l'application soit la suivante :

```
<Page>
  <StackPanel>
    <Rectangle Width=»200» Height=»50»>
      <Rectangle.Fill>
        <SolidColorBrush Color=»#FF432352» />
      </Rectangle.Fill>
    </Rectangle>
    <Rectangle Width=»200» Height=»50»>
      <Rectangle.Fill>
        <SolidColorBrush Color=»#FF432352» />
      </Rectangle.Fill>
    </Rectangle>
  </StackPanel>
</Page>
```

```
<ListView ItemsSource=»{Binding Personnes}»>
  <ListView.ItemTemplate>
    <DataTemplate>
      <Border>
        <Grid>
          <StackPanel>
            <TextBlock Text=»{Binding Nom}» Margin=»5» />
            <TextBlock Text=»{Binding Prenom}» Margin=»5» />
          </StackPanel>
        </Grid>
      </Border>
    </DataTemplate>
  </ListView.ItemTemplate>
</ListView>
```

On voit que la même couleur est utilisée à deux endroits et que deux pinceaux sont instanciés. On pourrait simplifier ce code et obtenir le même comportement en procédant de la manière suivante :

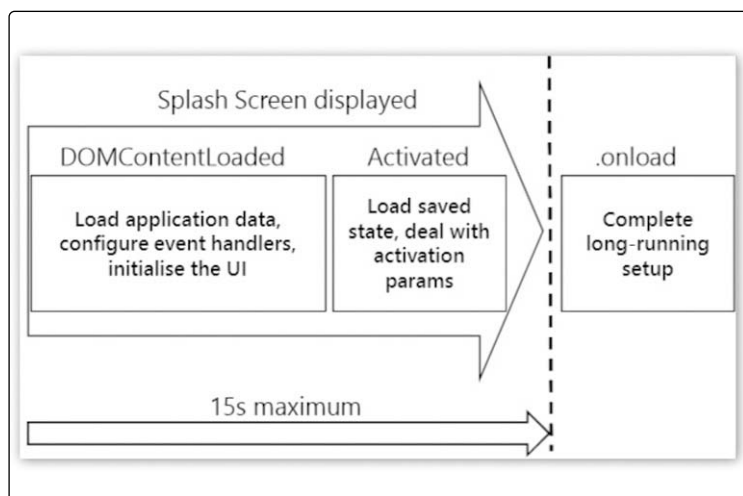
```
<Page>
  <Page.Resources>
    <SolidColorBrush x:Key=»myBrush» Color=»#FF432352» />
  </Page.Resources>
  <StackPanel>
    <Rectangle Fill=»{StaticResource myBrush}» Width=»200» Height=»50» />
    <Rectangle Fill=»{StaticResource myBrush}» Width=»200» Height=»50» />
  </StackPanel>
</Page>
```

Bien entendu, le cas précédent est trivial mais dans le cadre de la plupart des développements, on retrouve des éléments graphiques communs à divers endroits dans l'application.

Généralement ce sont les couleurs de fond, les tailles de textes, les polices de textes, etc. En les factorisant on gagne en mémoire, car les objets sont réutilisés, et en temps de parsing, car les fichiers à analyser sont de la sorte plus légers.

Supprimer les contrôles XAML inutiles

Une méthode plus longue mais aussi très efficace consiste à nettoyer le code XAML des contrôles superflus. Prenons l'exemple suivant :



Structure Splashscreen (wpfplayground.blogspot.fr)

Il y a ici un certain nombre de conteneurs inutiles. On peut supprimer la Border et la Grid sans changer le rendu visuel. Certains diront que ce n'est pas très important car ce ne sont que deux Panels. C'est la raison pour laquelle j'ai mis tout ceci comme ItemTemplate de la ListView. Supposons que l'on ait 100 personnes à afficher, cela reviendrait à une économie de 200 instanciations de contrôles. Sans compter le fait qu'à chaque mise à jour de la mise en page, le moteur de rendu appellera les méthodes MeasureOverride et ArrangeOverride de chaque Panel, ce qui constitue donc une économie de 400 appels de méthodes faisant un calcul de position. La suppression de ces deux contrôles est donc très rentable. Si vous avez ce genre de choses dans votre application vous savez maintenant quoi faire.

> Splashscreen étendu

Lorsque l'application nécessite beaucoup de chargements avant de pouvoir être opérationnelle on risque fortement de dépasser les 5 secondes allouées par le système de temps de démarrage. Malgré tout il y a certains scénarios où malgré toutes les optimisations possibles il sera impossible de charger l'application en moins de 5 secondes. Prenons l'exemple d'un jeu vidéo. Ce jeu devra probablement charger un certain nombre d'images, de sons et pourquoi pas diverses informations sur le Cloud avant de pouvoir être considéré comme opérationnel. Dans ce genre de scénario, on a la possibilité d'outrepasser cette limitation en créant un splashscreen étendu. Le concept du splashscreen étendu est simple. On va afficher une page très rapidement au démarrage de l'application montrant à l'utilisateur qu'un chargement est en cours. Pendant ce temps-là on effectue les traitements nécessaires et lorsque cela est fini on navigue sur la première page de l'application.

Un exemple typique de splashscreen étendu est un contrôle qui reprend l'image du Splashscreen et lui rajoute un ProgressRing en dessous du logo pour signifier à l'utilisateur que le chargement est en cours. Voici à quoi ressemblerait le contrôle ExtendedSplashScreen dans ce cas :

```
<Grid>
  x:Class=»Appl.ExtendedSplashScreen»
  xmlns=»http://schemas.microsoft.com/winfx/2006/xaml/presentation»
  xmlns:x=»http://schemas.microsoft.com/winfx/2006/xaml»>
  <Canvas>
    <Image x:Name=»image» Source=»Assets/SplashScreen.png» />
    <ProgressRing x:Name=»progressRing» Foreground=»White»
```



```
HorizontalAlignment=»Center» IsActive=»True» />
</Canvas>
</Grid>

public sealed partial class ExtendedSplashScreen
{
    public ExtendedSplashScreen(SplashScreen splash)
    {
        InitializeComponent();

        // On récupère les coordonnées du splashscreen pour placer l'image du
        // splashscreen étendu au même endroit
        image.SetValue(Canvas.LeftProperty, splash.ImageLocation.X);
        image.SetValue(Canvas.TopProperty, splash.ImageLocation.Y);
        image.Height = splash.ImageLocation.Height;
        image.Width = splash.ImageLocation.Width;

        // On positionne le ProgressRing juste en dessous
        progressRing.SetValue(
            Canvas.TopProperty,
            splash.ImageLocation.Y + splash.ImageLocation.Height + 32);
        progressRing.SetValue(
            Canvas.LeftProperty,
            splash.ImageLocation.X + (splash.ImageLocation.Width/2) - 15);
    }
}
```

Une des utilisations possibles ensuite de ce contrôle se fait dans l'évènement OnLaunched de la classe Application :

```
protected async override void OnLaunched(LaunchActivatedEventArgs args)
{
    SplashScreen splashScreen = args.SplashScreen;
    var eSplash = new ExtendedSplashScreen(splashScreen);

    Window.Current.Content = eSplash;
    Window.Current.Activate();

    // Chargement des données nécessaires ici
    await Task.Delay(3000);

    var rootFrame = new Frame();
    Window.Current.Content = rootFrame;
    Window.Current.Activate();

    if (rootFrame.Content == null && !rootFrame.Navigate(typeof(MainPage), args.Arguments))
    {
        // ...
    }
}
```



Fig.1

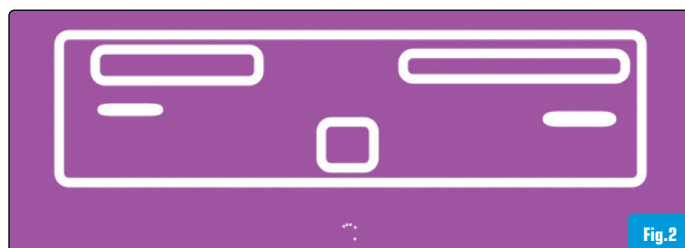


Fig.2

```
throw new Exception(«Failed to create initial page»);
}
```

Comme la plupart des splashscreen sont transparents on souhaite souvent que la couleur de fond du splashscreen étendu corresponde à celle du splashscreen. Dans ce cas il ne faut pas oublier de définir la couleur de fond du splashscreen dans le manifeste [Fig.1 et 2].

> Chargement des données

Le dernier grand axe d'optimisation présenté dans cet article est le chargement des données. La bonne gestion de cette partie est le plus souvent à traiter au cas par cas. Il faut se poser les questions suivantes pour chaque donnée à charger :

- Qu'est-ce que l'utilisateur veut voir en premier ?
- Dois-je afficher les données au fur et à mesure ?
- Comment montrer à l'utilisateur que des données sont en cours de chargement ?
- Si les données sont sur le Cloud puis-je maintenir un cache local ?

Charger en priorité ce qui doit être vu en premier

La taille d'un écran est limitée, on peut généralement jouer sur la vitesse de lancement perçue en chargeant au démarrage de l'application (ou dans le splashscreen étendu) la liste de ce que l'utilisateur souhaite réellement voir dès le premier affichage. Pour toutes les données secondaires on peut les charger en tâche de fond sans incidence sur l'expérience utilisateur. Procéder de la sorte permet de ne jamais présenter un écran vide car c'est, il faut l'avouer, fort désagréable.

Afficher des données au fur et à mesure

Il est généralement bon pour l'utilisateur d'afficher les données au fur et à mesure qu'elles arrivent. Cela lui donne l'impression que le traitement va plus vite. Dans ce cas on peut avoir à gérer une problématique d'ordonnancement des résultats. Rien n'assure en effet que les résultats en provenance du serveur seront dans l'ordre escompté et donc il faudra passer un peu de temps CPU à remettre tout cela dans le bon sens. A supposer que l'ordre ne soit pas important on pourrait transformer la méthode BulkLoad en UnorderedLoad pour avoir un code qui remonte les résultats dès que possible.

```
private async Task BulkLoad()
{
    var tasks = new Task<string>[5];
    for (int i = 0; i < 5; i++)
    {
        tasks[i] = DoWorkAsync(i);
    }

    await Task.WhenAll(tasks);

    foreach (Task<string> task in tasks)
    {
        string item = task.Result;
        BUListView.Items.Add(item);
    }
}

private async Task UnorderedLoad()
```

```

{
    var tasks = new Task<string>[5];
    for (int i = 0; i < 5; i++)
    {
        var task = DoWorkAsync(i);
        task.ContinueWith(    t => BUListView.Items.Add(t.Result),
TaskScheduler.FromCurrentSynchronizationContext());
        tasks[i] = task;
    }

    await Task.WhenAll(tasks);
}

```

Montrer à l'utilisateur que des données sont en cours de chargement

Il existe deux contrôles à notre disposition pour cela. La ProgressBar et le ProgressRing. La ProgressBar est la cousine de ce que l'on trouve sur Windows Phone. C'est une animation comportant une série de points qui se suivent de manière très fluide sur une ligne. L'avantage de ce contrôle est qu'il utilise peu de place et qu'on peut le mettre tout en haut de la page comme sur Windows Phone. C'est discret, simple à mettre en place et à utiliser. De plus, elle dispose de deux modes de fonctionnement. En mode déterminé on va devoir préciser une valeur minimum et maximum au contrôle (qui correspondront aux extrémités du contrôle) et la valeur courante. Ce mode est utile lorsque par exemple on charge 20 fichiers et que l'on veut montrer à l'utilisateur le nombre de fichiers déjà chargés. En mode indéterminé les points constituant la barre de progression tourneront en boucle pour informer l'utilisateur qu'une opération est en cours mais sans précision quant à sa progression. Le ProgressRing, lui, n'a qu'un fonctionnement indéterminé. Son animation est elle aussi constituée de points mais cette fois ils forment un cercle et prennent donc plus de place (Voir [Fig.1](#)).

Dans le cas d'un fonctionnement indéterminé la ProgressBar (voir [Fig.2](#)) a tout son sens lorsque l'écran est déjà en partie rempli et qu'il faut donc informer l'utilisateur sans prendre trop d'espace à l'écran. Au contraire, le ProgressRing prend tout son sens si l'écran, ou la partie de l'écran où sont chargées les données, est vide. Il permet d'occuper l'espace et évite donc à l'utilisateur une sensation de vide. Voici comment s'utiliserait une ProgressBar en mode indéterminé :

```

<Page>
    <Grid Background="{StaticResource ApplicationPageBackground
ThemeBrush}">
        <ListView x:Name="BUListView" />
        <ProgressBar x:Name="progressBar" VerticalAlignment="Top"
IsIndeterminate="False" Visibility="Collapsed" />
    </Grid>
</Page>

private async Task UnorderedLoad()
{
    progressBar.Visibility = Visibility.Visible;
    progressBar.IsIndeterminate = true;

    var tasks = new Task<string>[5];
    for (int i = 0; i < 5; i++)
    {

```

```

        var task = DoWorkAsync(i);
        task.ContinueWith(    t => BUListView.Items.Add(t.Result),
TaskScheduler.FromCurrentSynchronizationContext());
        tasks[i] = task;
    }

    await Task.WhenAll(tasks);

    progressBar.IsIndeterminate = false;
    progressBar.Visibility = Visibility.Collapsed;
}

```

```

Nom : Dupont
Nom : Li
Nom : Matsumoto
Nom : Smith

```

Maintenir un cache local de données du Cloud

Dans la mesure du possible c'est très conseillé ! Il est beaucoup plus rapide de charger des données depuis un fichier en local que depuis un service web. Dès lors, au démarrage on va charger les anciennes données depuis le cache local et lancer un rafraîchissement de ce cache dans une tâche de fond. Cette tâche mettra aussi à jour l'interface graphique pour afficher les changements. L'exemple par excellence pour cela est une application dont la fonction est d'afficher des articles (blog, presse ou autres). On garde en cache les articles chargés auparavant, on les affiche dès le lancement depuis l'application, on met une ProgressBar en mode indéterminé et on rafraîchit la liste des articles une fois que l'application est lancée. L'avantage de procéder ainsi est que l'on reste dans nos 5 secondes d'initialisation (voire un peu plus avec un Splashscreen étendu), tout en offrant à l'utilisateur du contenu très rapidement. Là encore, on effectue côté CPU plus de travail car on charge depuis un cache local et depuis le Cloud mais l'utilisateur perçoit une application plus rapide.

> Conclusion

Dans cet article j'ai souvent passé mon temps à vous expliquer comment perdre du temps pour en gagner ! C'est là tout le paradoxe entre temps de chargement effectif et perçu. Il est facile pour un développeur de mesurer le temps effectif mais très difficile d'évaluer le temps perçu. Le moyen le plus simple consiste à faire tester votre application à d'autres que vous. Vous pouvez aussi tenter de vous replacer en tant qu'utilisateur lorsque vous effectuez vos tests mais ce n'est pas aussi facile qu'il y paraît. Si à la fin de cet article vous avez retenu qu'il faut toujours montrer des données aux utilisateurs de vos applications et les informer lorsque vous faites de longs traitements, alors vous leur rendrez service. Vous vous rendrez service à vous aussi car ils seront probablement beaucoup plus enclins à utiliser régulièrement votre application.

John Thiriet

Expert XAML (Windows 8, Windows Phone, WPF, Silverlight)

MCNEXTE (www.mcnext.com)

<http://blog.john-thiriet.com>

Migrer une application Windows Phone 7 vers Windows 8

Si vous êtes familier avec Windows Phone, vous avez pratiquement toutes les cartes en main pour développer des applications Windows 8. Les changements entre les deux plateformes sont simples mais assez nombreux. Nous ne les verrons pas tous dans cet article, mais les principaux, comme l'interface utilisateur, les inputs, les tuiles, le système de fichiers ou encore les principaux capteurs.

> Prérequis techniques pour bien commencer

Avant de porter votre application il est nécessaire de faire en sorte que la logique de votre application soit détachée de l'interface et des appels API (système de fichiers, etc.).

Vous pouvez utiliser pour cela un design pattern très répandu dans l'univers XAML, le Model-View-ViewModel (MVVM). Il permet de séparer l'interface utilisateur de la logique de votre application. Pour résumer, le Model correspond à la communication/récupération des données brutes (C#), le ViewModel contient la logique (mise en forme des données pour la vue, système de recherche dans une liste, etc.) (C#) et la View est l'interface utilisateur (XAML).

Une fois ce model mis en place vous pourrez partager les fichiers (correspondant à la logique de votre application) entre vos projets Windows Phone 7 et Windows 8. Attention toutefois, il est possible que dans vos fichiers de logique vous ayez quelques lignes spécifiques, pour cela pensez à utiliser les conditions de compilation (#if WINDOWS_PHONE ... #elif WINDOWS8 ... #endif). Le meilleur moyen pour partager ses fichiers est la Portable Class Library. C'est un type de class library permettant d'avoir le même code portable sur quatre plateformes (Windows 8 Apps, Windows Phone, Silverlight et WPF).

> L'Interface Utilisateur

Windows 8 et Windows Phone sont tous deux basés sur les mêmes principes d'UI appelés Modern UI (idem pour l'expérience utilisateur appelée Modern UX).

Les tailles d'écrans et les interactions n'étant pas les mêmes, il est évident que les contrôles sont différents, les informations n'étant

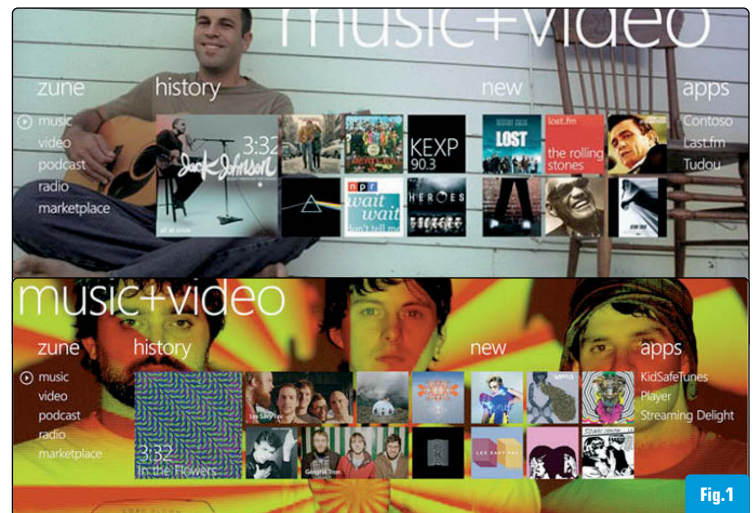


Fig.1

pas affichées de la même manière. La migration de l'UI est simple à partir du moment où vous savez à quoi correspondent les contrôles Windows 8 par rapport aux contrôles Windows Phone. Par exemple, le Panorama, un des contrôles les plus utilisés sous Windows Phone. Il est équivalent à une couverture de magazine, on peut y voir les trois ou quatre dernières news, la personnalité/musique du moment mise en avant, etc. Si l'on résume techniquement, le Panorama contient peu de données mais permet de les mettre en avant. Sous Windows 8 nous retrouvons ces principes, le Panorama est devenu la GridView. Elle est une suite de groupes d'éléments, ce qui ressemble énormément à un Panorama déplié (équivalent de

l'écran de démarrage avec les tuiles d'applications). On reste donc dans le thème du magazine (où chaque groupe représente une information avec seulement quelques éléments mis en avant) [Fig.1 et 2].

> La Navigation

Windows 8 a gardé le principe de navigation de Windows Phone (qui l'avait lui-même gardé de Silverlight). Cependant un léger changement arrive. Sous Windows Phone la navigation se fait

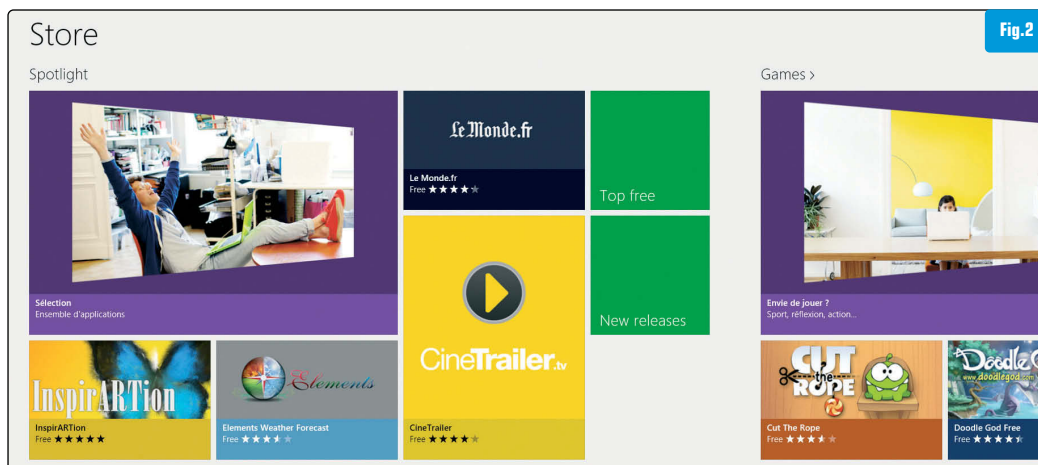


Fig.2

à base de chaînes de caractères, alors que sous Windows 8 elle s'effectue à l'aide d'un type. Second point, qui n'est pas négligeable, les paramètres que l'on souhaite passer à cette nouvelle page se font tout simplement par un objet. Il n'est donc plus nécessaire d'avoir sa classe de navigation personnalisée afin de passer simplement des arguments.

Windows Phone 7

```
NavigationService.Navigate(new Uri("/ProductPage.xaml?Id=5", Uri
Kind.Relative));
```

Windows 8

```
Frame.Navigate(typeof(ProductPage), myId);
```

> Les Inputs

Les inputs ont changé, et cela pour le plus grand plaisir des développeurs. Sur Windows Phone cela était assez simple, les téléphones n'ayant que du tactile, il suffisait de s'abonner aux événements souris (MouseLeftButtonDown, ...) ou simplement Tap, il n'y avait aucun souci. Mais Windows 8 n'est pas que tactile, la souris et d'autres systèmes d'inputs peuvent être également utilisés afin de pointer un endroit de l'écran. Sur une application utilisant le tactile et la souris en WPF, on doit utiliser les événements tactiles et souris, mais lorsque l'on est sur un écran tactile les deux événements peuvent être lancés... Pour éviter cela sur Windows 8, Microsoft a ajouté une nouvelle gestion des inputs appelée Pointer, ce qui permet d'éviter ce problème et de regrouper tous ces types d'inputs.

Sur des contrôles basiques (comme les boutons) cela donnera donc :

Windows Phone

```
MouseLeftButtonUp=>eventhandler»
```

Windows 8

```
PointerReleased=>eventhandler»
```

Autre point intéressant, l'apparition d'ItemClick sur les contrôles ayant des collections (comme la GridView). Cela permet d'avoir un seul événement à surveiller au lieu d'en avoir un par élément.

> Les Tuiles

Les Tuiles étaient la grande nouveauté de Windows Phone 7. Pas mal d'évolutions pour celles-ci sous Windows 8. On peut, bien évidemment, toujours avoir une tuile principale ainsi que des tuiles secondaires permettant d'arriver directement à un point donné dans l'application.

Il y a principalement 3 changements concernant les tuiles. Le premier est que l'on peut créer de petites et de grandes tuiles (carrées et rectangulaires), chose qui n'était pas

possible sous Windows Phone 7. La seconde nouveauté se situe au niveau de la création des tuiles secondaires. Lors de la création de tuiles sous Windows Phone il fallait utiliser la classe ShellTileData en remplissant les champs de base tels que Title, Count, BackgroundImage, etc. Cela donnait des tuiles simples à faire mais toutes identiques (au détail près de l'image de fond). Pour avoir une tuile différente il fallait passer par un processus complexe de création d'image personnalisée en code (ou par un serveur permettant de la mettre à jour. Tout cela est fini sous Windows 8 ! Les tuiles sont sous format XML, et Microsoft fournit pas moins de 46 modèles de tuiles ! Afin de faciliter le remplissage du XML, la classe TileContentFactory est fournie dans le code d'exemple appelé App tiles and badges sample.

Le troisième changement concerne une nouveauté. Un nouvel élément vient enrichir les tuiles : les badges. Si vous ne les avez pas encore vus, ce sont de petites icônes présentes en bas à droite des tuiles qui en possèdent. Les badges sont des icônes prédéfinies (donc impossible de mettre un badge personnalisé).

Windows Phone

```
tile = ShellTile.ActiveTiles.FirstOrDefault(t => t.Navigation
Uri.ToString().Contains(«tilenumber=2»));
if (tile != null)
{
    tileData = new StandardTileData
    {
        Title = «New Title»
    }
    tile.Update(tileData);
}
```

Windows 8

```
rectangle = TileContentFactory.CreateTileWideImageAndText01();
square = TileContentFactory.CreateTileSquareImage();
rectangle.SquareContent = squareContent;

rectangle.TextCaptionWrap.Text = «Image and text Tile»;
```

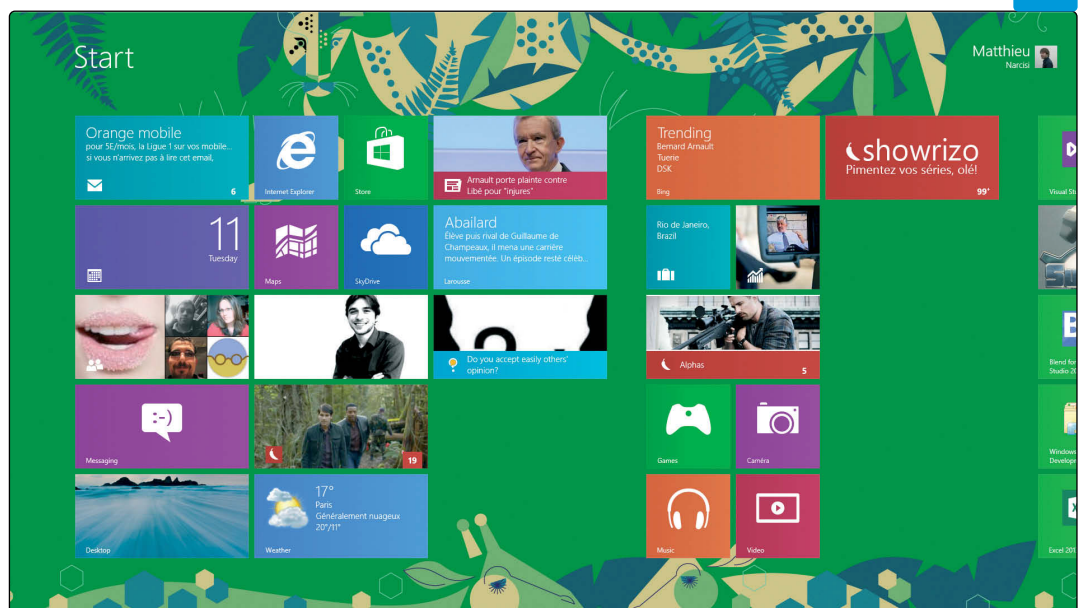


Fig.3

```
rectangle.Image.Src = «ms-appx:///images/Rectangle.png»;
square.Image.Src = «ms-appx:///images/Square.png»;
```

```
TileUpdateManager.CreateTileUpdaterForSecondaryTile(id).Update
(rectangle.CreateNotification());
```

[Fig.3]

> Les Toasts

Pour les toasts, bonne nouvelle, ils ont suivi les tuiles ! Il y a 7 modèles de toasts différents, basés sur du XML tout comme les tuiles. Une classe permettant de les remplir aisément existe également, elle a pour nom `ToastContentFactory`.

Windows Phone

```
ShellToast toast = new ShellToast
{
    Title = «Title»,
    Content = «Content»,
    NavigationUri = new Uri(«ProductDetail.xaml?Id=42», Uri Kind.
Relative)
};
toast.Show();
```

Windows 8

```
toast = ToastContentFactory.CreateToastImageAndText01();
toast.TextHeading.Text = «Heading »;
toast.TextBodyWrap.Text = «Body»;
toast.Launch = «Id=42»;
toast.Image.Src = «ms-appx:///images/toastImage.png»;

ToastNotificationManager.CreateToastNotifier().Show(toast.Create
Notification());
```

> Paramètres Locaux (Settings)

Les settings n'ont pas tellement changé sur le principe, mais il est possible sur Windows 8 de créer différents conteneurs. Ils permettent d'organiser les paramètres de l'application. À noter que la profondeur des conteneurs est restreinte à 32 niveaux (ce qui est bien suffisant je vous l'accorde).

> Système de fichiers

Un des principes fondamentaux des applications Modern UX est d'être rapides et fluides. L'accès au système de fichiers a donc été revu. Sous Windows Phone c'est le développeur qui devait faire en sorte que l'accès à un fichier ne soit pas bloquant pour l'application. Mais Windows 8 a rendu les appels au système de fichiers entièrement asynchrones.

Mais pas n'importe comment, pas question de s'encombrer d'une callback à chaque appel. Deux nouveaux mots clés ont fait leur apparition : `async` et `await`. Ils permettent de ne plus utiliser de callback pour les appels asynchrones. Pour plus de détails, voir l'article Les nouveautés de Windows 8 pour les développeurs (Programmez ! 156).

Windows Phone

```
using (var storage = IsolatedStorageFile.GetUserStoreForApplication())
{
    storage.CreateDirectory(«NewDirectory»);
```

```
File = store.CreateFile(«File.xml»);
File.Close();
}
```

Windows 8

```
folder = ApplicationData.Current.LocalFolder;
file = await folder.CreateFileAsync(«AsyncFile.txt», Create
CollisionOption.ReplaceExisting);
await FileIO.WriteTextAsync(file, «Answer to the Ultimate
Question of Life, the Universe, and Everything»);
```

> Capteurs (Localisation, Accéléromètre et Photo)

Windows 8 étant destiné également aux tablettes, Microsoft se devait de mettre à disposition dans son API une façon de récupérer sa position GPS. Un léger changement comparé à Windows Phone, les développeurs n'ont plus à « lancer » la géolocalisation. Il suffit d'instancier une classe et de s'abonner à l'évènement de changement de position.

Windows Phone

```
watcher = new GeoCoordinateWatcher();
watcher.PositionChanged += positionChanged;
watcher.TryStart(false, TimeSpan.FromMilliseconds(5000));

private void positionChanged(object sender, GeoPositionChanged
EventArgs<GeoCoordinate> e)
{
}
```

Windows 8

```
locator = new Geolocator();
locator.PositionChanged += positionChanged;

private void positionChanged(Geolocator sender, PositionChanged
EventArgs e)
{
}
```

L'accéléromètre quant à lui n'a pas changé, mis à part le fait qu'il est possible sous Windows 8 que votre machine n'en possède pas (un ordinateur fixe par exemple). Une vérification est donc nécessaire :

```
accelerometer = Accelerometer.Default();
if (accelerometer != null)
{
    accelerometer.ReadingChanged += ReadingChanged;
}
```

Comme vous avez pu le constater, les changements sont assez simples, la plus grande difficulté réside dans le fait de garder sa logique séparée de l'interface utilisateur et du système de fichiers. Pour une liste complète des changements entre Windows Phone et Windows 8, rendez-vous sur MSDN (<http://msdn.microsoft.com/en-us/library/windows/apps/hh465136%2Bv=VS.85%29.aspx>).

Benjamin Baldacci
Consultant .Net à Wygwam

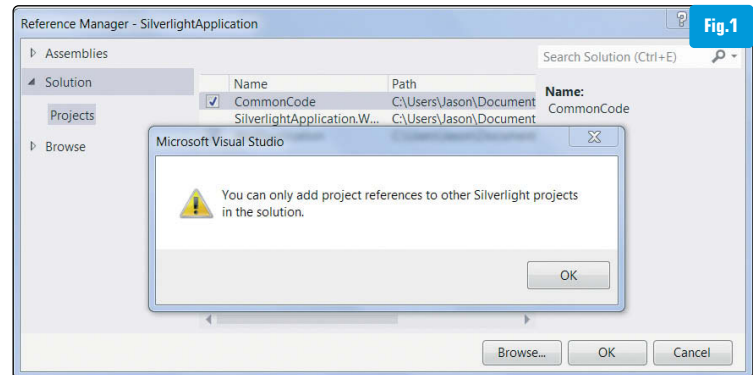
Portable Class Library avec Visual Studio 2012

Avec la multiplication des supports (PC, tablettes, Smartphones, etc.), et des technologies de l'écosystème Microsoft (WPF, Silverlight, Windows Phone, Windows Azure, Xbox 360, etc.), la compatibilité des composants entre les technologies devient un sujet incontournable pour obtenir des développements de qualité. Il s'agit là de ne pas avoir à dupliquer des composants et donc du code pour chaque plateforme utilisée. Microsoft adresse ce sujet dans sa nouvelle version de Visual Studio 2012 avec la « Portable Class Library ».

Avertissement :

ce n'est pas du Windows 8, mais du pur Visual Studio 2012

Lorsque Silverlight est apparu, beaucoup de développeurs ont été confrontés à la même problématique : comment partager et mutualiser les développements communs entre des applications basées sur le framework .NET standard et celles basées sur le framework .NET Silverlight. Car, par défaut, il n'est pas possible de mélanger des bibliothèques .NET standard et Silverlight et de les utiliser dans un même projet. Si on essaie cela, une erreur apparaît [Fig.1].

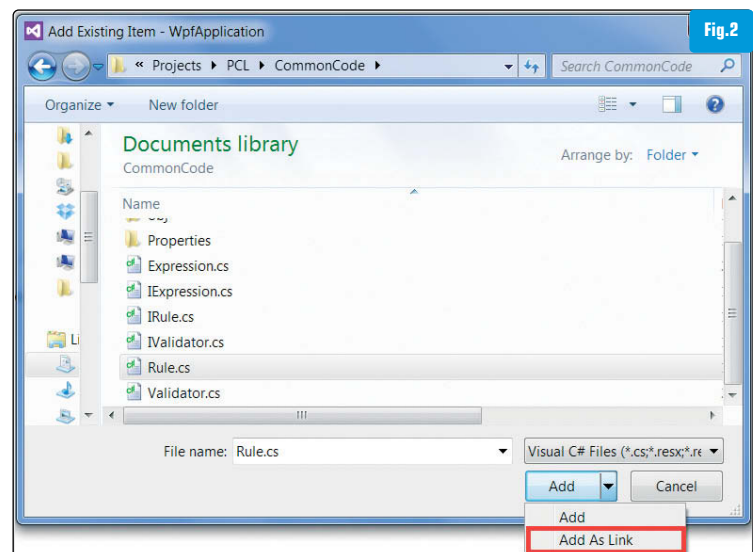


> Solution de contournement : Add as Link

A l'époque, la solution la plus répandue et aussi préconisée par Microsoft était plutôt une solution de contournement qu'une solution pérenne. Il s'agissait de dupliquer les bibliothèques (au moins une par plateforme) sans dupliquer les fichiers sources : pour cela, il fallait rajouter les fichiers sources existants via la fonctionnalité « Add as Link » intégrée dans Visual Studio dans la boîte de dialogue « Add Existing Item » [Fig.2]. Après avoir compilé, on constate que le fichier n'a pas été copié vers le projet, mais à la place une référence y a été ajoutée.

Les fichiers sources ont donc été partagés, toutefois sans duplication, mais pas les bibliothèques : il existe bien une version de bibliothèque qui doit être compilée pour chaque plateforme visée.

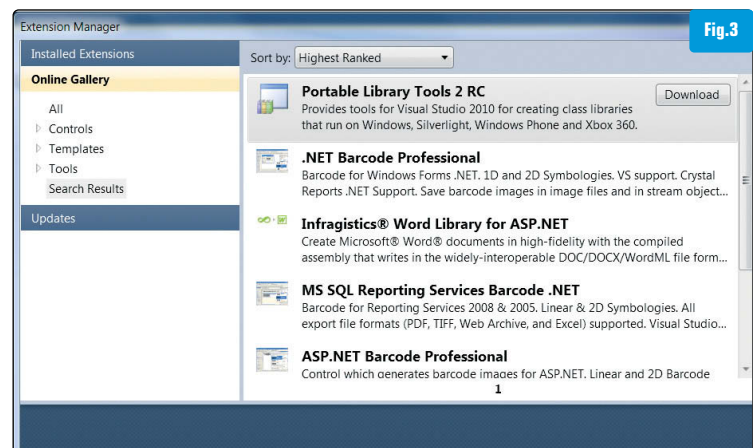
Cette solution n'était bien évidemment pas satisfaisante, de plus la multiplication des plateformes dans les développements modernes rend cette approche laborieuse et inapplicable.



> Solution via NuGet package : Portable Library Tools

Microsoft, ne pouvant laisser cette situation en l'état, s'est attaqué à ce problème et a sorti une extension à Visual Studio 2010 SP1 permettant d'écrire des bibliothèques utilisables par plusieurs plateformes via les « Portable Library Tools » actuellement en version 2 RC disponible ici : <http://visualstudiogallery.msdn.microsoft.com/b0e0b5e9-e138-410b-ad10-00cb3caf4981/>.

Encore plus facile, ce package peut aussi être téléchargé et installé directement via NuGet depuis l'« Extension Manager » de Visual Studio 2010 [Fig.3]. Une fois ce package installé, il est possible de créer un nouveau projet de type « Portable Class Library ». Ce projet permet de générer des bibliothèques pouvant être utilisées directement par plusieurs plateformes qui doivent toutefois être choisies au préalable. La première version de cette extension a été mise en ligne en janvier 2011 et mise à jour régulièrement pour pouvoir prendre en compte :



- Les nouvelles plateformes comme Modern UI (Windows 8), Windows Phone 7.5, Silverlight 5, etc.
- Les nouvelles versions du framework .NET comme la version .NET 4.5
- Ou bien encore pouvoir supporter de nouvelles fonctionnalités comme Code Contract

> Solution intégrée à Visual Studio 2012 : Portable Class Library (PCL)

Désormais, le type de projet « Portable Class Library (PCL) » est directement intégré à Visual Studio 2012 et c'est une bonne nouvelle, car les exigences en termes de qualité, maintenabilité et fonctionnalités se sont accrues.

La PCL peut s'avérer très utile pour partager des algorithmes (par exemple pour la validation des règles métier), des interfaces ou des objets de données entre plusieurs applications. Comme déjà évoqué, ces applications peuvent être basées sur des technologies différentes. Un autre cas d'utilisation intéressant est le stockage du View-Model dans une librairie PCL quand on souhaite implémenter le pattern MVVM [Fig.4]. Le View-Model peut alors être partagé dans toutes vos applications (Windows, Web, Mobile, etc.).

L'utilisation de la PCL permet donc la consistance et l'encapsulation du code indépendamment de l'endroit où il sera utilisé. Ceci réduit les coûts de maintenance et augmente la productivité de l'équipe. Voici un tableau avec les fonctionnalités supportées par technologie en utilisant la PCL :

Fonctionnalité	.NET Framework	Modern UI style	Silverlight	Windows Phone	Xbox 360			
	4	4.0.3	4.5		4	5	7	7.5
Core BCL	X	X	X	X	X	X	X	X
Core XML	X	X	X	X	X	X	X	X
LINQ	X	X	X	X	X	X	X	X
IQueryable	X	X	X	X	X	X		X
Dynamic Keyword			X	X	X	X		
Core WCF	X	X	X	X	X	X	X	X
Core Networking	X	X	X	X	X	X	X	X
MEF	X	X	X	X	X	X		
Data Contract								
Serialization	X	X	X	X	X	X	X	X
XML Serialization	X	X	X	X	X	X	X	X
Json Serialization	X	X	X	X	X	X	X	X
View models	X	X	X	X	X	X		
Data annotations	X	X	X	X	X			
XLINQ	X	X	X	X	X	X	X	X
System.Numerics			X	X	X	X	X	X

Il faut noter que la PCL permet seulement l'utilisation d'un ensemble de fonctionnalités communes aux différentes technologies : on est donc contraint de cibler le plus petit dénominateur commun. Ceci est compréhensible, puisque toutes les fonctionnalités ne sont pas forcément disponibles dans chaque technologie. Dans la suite de cet article, nous allons tenter de démystifier la PCL par le biais d'un exemple simple qui couvrira un maximum de plateformes.

Création du projet :

Intéressons-nous maintenant à l'utilisation de la PCL avec Visual Studio 2012. La première chose à faire est de créer un nouveau projet de type PCL [Fig.5].

Ensuite, vous devez sélectionner les frameworks cibles avec lesquels votre librairie PCL doit être compatible. Cette étape est très importante, car plus vous allez en ajouter, moins vous aurez de fonctionnalités accessibles : si vous souhaitez développer un composant qui doit être utilisé par des projets qui utilisent les framework .NET 3.5 et .NET 4.5 par exemple, vous n'aurez pas accès aux fonctionnalités liées à « Code Contract ».

Notez qu'une fois compilée, la librairie PCL qui en résulte peut être référencée sans aucune modification ni configuration.

Une autre information intéressante est que vous avez la possibilité de modifier (ajouter et supprimer) la liste des technologies supportées après la génération du projet. Une bonne pratique consiste donc à inclure, dans un premier temps, uniquement les frameworks nécessaires et à ajouter les autres au fur et à mesure que les besoins se présentent.

Targeting Packs :

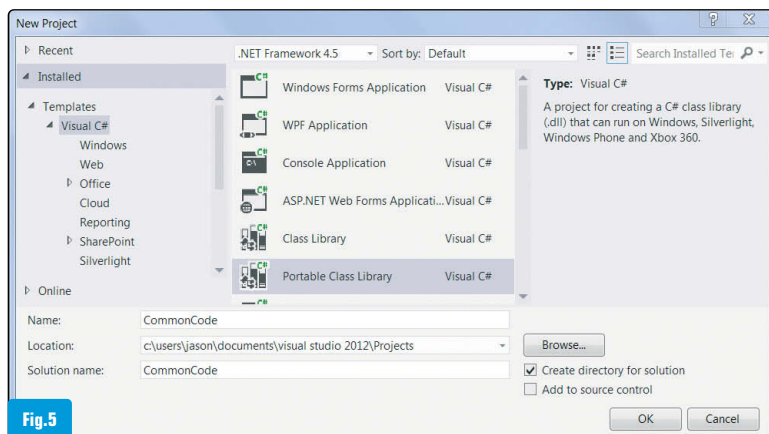
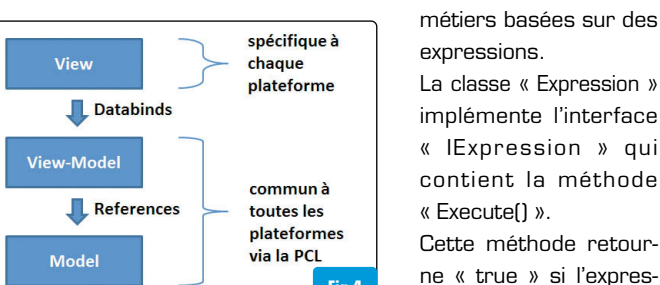
Vous avez également la possibilité d'étendre le nombre de frameworks supportés en téléchargeant d'autres packs de frameworks pour Visual Studio 2012. Ils peuvent contenir des mises à jour pour Visual Studio et doivent être installés avec les runtimes correspondants.

Voici les technologies supportées et les packs actuellement disponibles au téléchargement (<http://msdn.microsoft.com/en-us/hh487283.aspx>) :

- .NET Framework 4.5 (inclus dans Visual Studio 2012)
- .NET Framework 4.0.3, 4.0.2, 4.0.1
- .NET Framework 4 (inclus dans Visual Studio 2012)
- .NET Framework 2.0/3.0/3.5 SP1
- Windows Azure (via Windows Azure SDK)
- Xbox 360 (via XNA Game Studio 4.0)
- Silverlight 5 (inclus dans Visual Studio 2012)
- Windows Phone 7.5 (inclus dans Visual Studio 2012)
- Modern UI style / Windows 8 (inclus dans Visual Studio 2012)

Implémentation :

Prenons un exemple concret dans Visual Studio 2012 pour implémenter des fonctionnalités d'un validateur de règles métiers basées sur des expressions.



sion a abouti sans erreur, sinon elle retourne « false ». Dans cet exemple simple, elle retournera toujours « true ».

```
public interface IExpression
{
    bool Execute();
}
public class Expression : IExpression
{
    public bool Execute()
    {
        // Do something
        //return true if sucessful
        //return false if not sucessful
        return true;
    }
}
```

La classe « Rule » implémente l'interface « IRule » qui contient deux propriétés : « RuleId » du type unsigned integer et « ExpressionList » du type IExpression (défini ci-dessus). Cette classe a pour objectif de stocker toutes les expressions pour une règle métier.

```
public interface IRule
{
    uint RuleId { get; }
    IList<IExpression> ExpressionList { get; }
}
public class Rule : IRule
{
    private uint _ruleId;
    private IList<IExpression> _expressionList = new List<IExpression>();

    public Rule(uint ruleId)
    {
        _ruleId = ruleId;
    }

    public uint RuleId
    {
        get { return _ruleId; }
    }

    public IList<IExpression> ExpressionList
    {
        get
        {
            return _expressionList;
        }
    }
}
```

La classe « Validator » implémente l'interface « IValidator » qui contient la méthode « ValidateRules(...) ». Cette méthode a comme paramètre d'entrée une liste de règles métiers du type « IList<IRule> » (défini ci-dessus). Elle retourne « true » si la validation de toutes les expression a abouti sans erreur, sinon elle retourne « false ».

```
public interface IValidator
{
    bool ValidateRules(IList<IRule> ruleList);
}
public class Validator : IValidator
{
    public bool ValidateRules(IList<IRule> ruleList)
    {
        bool result = true;

        foreach (var rule in ruleList)
        {
            foreach (var expression in rule.ExpressionList)
            {
                result &= expression.Execute();
            }
        }

        return result;
    }
}
```

Une fois le code compilé, vous pouvez utiliser cette librairie depuis n'importe quel framework sélectionné au début. Voici un exemple d'implémentation du validateur de règles métiers :

```
IExpression expression1 = new CommonCode.Expression();
IExpression expression2 = new CommonCode.Expression();

IRule rule = new Rule(1);
rule.ExpressionList.Add(expression1);
rule.ExpressionList.Add(expression2);

var ruleList = new List<IRule>();
ruleList.Add(rule);

IValidator validator = new Validator();
bool isValid = validator.ValidateRules(ruleList);
```

Cet exemple simple nous a permis de voir comment partager des interfaces, des classes et du code métier dans une librairie PCL unique.

Conclusion

Microsoft fait des progrès dans l'unification du développement multiplateforme en fournissant un moyen simple de partager du code via la Portable Class Library (PCL). De plus, même Mono 2.12 (<http://www.mono-project.com>), Mono for Android (<http://xamarin.com/monoforandroid>) et MonoTouch pour iPhone (<http://xamarin.com/monotouch>) supportent maintenant la PCL ce qui permet un partage de code encore plus large.



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



Fathi Bellahcene
Software Architect MVP C#.Net Rangers by Sogeti
Blog : <http://blogs.codes-sources.com/fathi>

Testeur : un métier réellement d'avenir

Le testeur est un métier qui fait maintenant partie du paysage informatique et du quotidien des développeurs. Mais est-ce un métier identifié dans une SSII ou une équipe IT ? En fait, le terme « testeur » veut tout dire et ne rien dire, car il recouvre une réalité très diverse. Mais, c'est un profil prometteur qui ne faut pas négliger, même si en France ce rôle a encore du mal à être parfaitement reconnu.

Si nous devons donner une définition basique du testeur : c'est celui qui met en place des batteries de tests et les applique pour tester et qualifier le code ou un logiciel selon différents types.

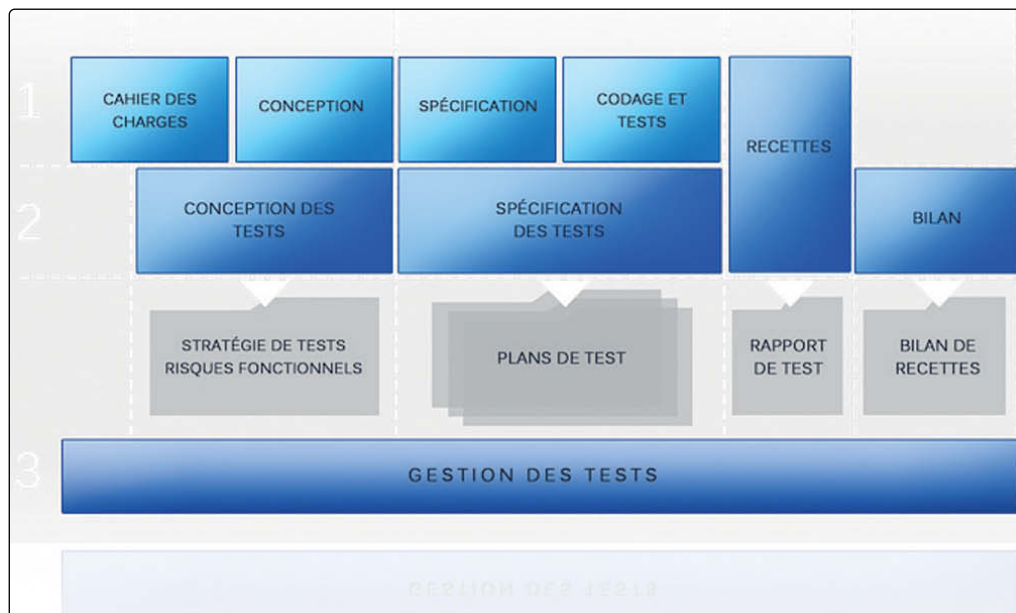
Le CFTL (Comité Français du Test Logiciel) donne une nomenclature très complète du métier et illustre parfaitement la diversité du testeur (voir encadré).

> Un métier manquant toujours de visibilité ?

Pôle Emploi répertorie dans le code ROME 1805 le métier de testeur informatique. Mais bien souvent, dans les annonces d'emploi, testeur sera intégré à l'intérieur d'une dénomination plus large : développeur, ingénieur informatique. Ainsi à la date du 17 octobre, 13 offres d'emploi « testeur informatique » étaient intitulées « testeur ». Les Jeudis, à la même date, référençaient seulement 8 annonces ! « *Le métier de testeur est visible sur le marché et est en hausse sur la qualité et le test, malgré les économies budgétaires et les coûts en baisse.* » précise **Natasha Pickaert** (National Practice Manager Testing chez Sogeti France). Il est très



difficile de connaître le nombre de testeurs (selon la nomenclature CFTL) qui sont en poste en France. Le profil va du débutant à l'expert sénior (ingénieur de test, chef de projet) et la rémunération fait donc l'élastique : de 18 / 20 000 € à 50 000 € (annuel, brut). Pour un testeur débutant, la fourchette est celle du développeur : entre 20 - 30 000 €. La grille salariale est la même que pour le développeur, l'ingénieur



“Une rémunération annuelle de 18 000 à 50 000 euros”

en informatique. Le profil va changer d'une entreprise à une autre. Par exemple, chez Sogeti, les candidats testeurs seront plutôt des bac + 4 ou 5 et non des bac + 3.

L'expérience joue beaucoup, même sur un poste identique, et bien entendu les compétences et la maîtrise des outils et des bonnes pratiques, avec une forte compétence en management et organisation. Bien souvent, on débute par être « simple » testeur pour ensuite monter en compétence et en responsabilité. Pour Natasha Pickaert, c'est un métier qui exige méthode et rigueur. Il faut être capable de rédiger avec précision les anomalies pour aider le développeur à les trouver, les comprendre et les corriger, puis on organise la qualité au niveau de l'entreprise, ensuite, restent à définir les tests et les méthodes de tests à mettre en œuvre.

Les testeurs se retrouvent dans des SSII

World Quality Report 4e édition

L'étude (conjointe à Sogeti, Cap Gemini et HP) révèle également que plus des deux cinquièmes (42 %) des budgets consacrés à l'assurance qualité ont augmenté l'année dernière et que 11 % seulement d'entre eux ont diminué, car les entreprises cherchent à réduire les coûts et à accélérer la mise à disposition des services. La majorité des entreprises (53 %) prévoient également d'augmenter leur budget dédié à l'assurance qualité d'ici 2015, tandis que seulement un cinquième (18 %) pensent le réduire. L'étude indique aussi que les entreprises qui s'efforcent de centraliser et de consolider leurs processus d'assurance qualité pour réduire encore leurs coûts et générer des gains d'efficacité ont de plus en plus tendance à investir dans des « Centres d'Excellence » dédiés au Testing. Actuellement, 6 % des entreprises seulement disposent d'un Centre d'Excellence pleinement opérationnel, mais ce chiffre représente une augmentation de 50 % par rapport à 2011 et près de deux tiers des entreprises (60 %) en construisent un ou prévoient de le faire pour renforcer leur compétitivité.

(spécialisées testing ou généralistes) et dans les entreprises. Il peut être, à l'origine, technique ou purement métier. Pourtant, le testing connaît une croissance très dynamique, par rapport au reste du secteur IT : + 10 % en moyenne par an jusqu'en 2015 (étude PAC, juin 2012). Le marché français rattrape, lentement, son retard sur les autres pays comme l'Allemagne, l'Angleterre, qui sont deux marchés structurés et matures. « L'activité du Test s'est lentement structurée pour devenir un métier à part entière dans la filière informatique, avec des formations et certifications dédiées mais aussi des parcours de carrière et des passerelles métiers. Ce travail de promotion du testing a permis de réduire les fractures structurelles du marché qui le pénalisaient au niveau des ressources compétentes », explique Arnold Aumasson, Senior Vice-Président et expert Testing chez Pierre Audoin Consultants (PAC).

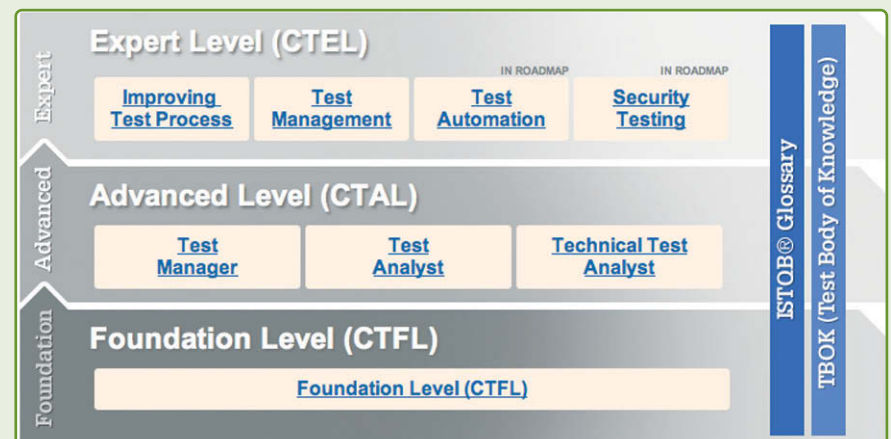
Comme nous l'avons dit plus haut, les postes se trouvent aussi bien en entreprise que dans les SSII. Parmi ces dernières, certaines sont des SSII testing c'est-à-dire spécialisées dans les tests. Il en existe plusieurs en France : Scopteam, Open SSII test, Acial (plus orientée conseil), Pentalog... Ces acteurs seront soit nationaux, soit régionaux.

> Formation et certification

La majorité des testeurs n'ont pas suivi de cursus spécifique testing dans une école d'ingénieur informatique ou à l'université. Le profil initial du testeur n'est pas uniforme. Il en est arrivé là par des formations dédiées, une veille technologique et au fur et à mesure des projets. La veille technologique est très importante pour pouvoir monter en compétence car le domaine évolue régulièrement dans les outils, les méthodes de tests et la manière de les structurer et de définir les campagnes. Les SSII forment en interne les futurs testeurs. Une certification reconnue est un « plus » indéniable sur son CV. La plus connue est celle du CFTL (qui comprend différents niveaux de compétences et de connaissances). Sur les 9 premiers mois de 2012, 800 testeurs ont été certifiés.

Sur les cursus, il existe des spécialisations, ou des approfondissements, à Angers (QUASSI), l'université de Franche-Comté ou encore à Rennes et l'école des Mines d'Alès. La France souffre toujours d'un manque de cursus spécifiques contrairement à l'Allemagne ou à l'Angleterre.

Les métiers du test (source CFTL)



- **Testeur (Bac +3)** : souvent débutant, il est dans l'entreprise. Il exécute des tests conçus par des tierces personnes. Il remplit des fiches d'anomalies, un bugtrack. Il utilise un outil de gestion des tests.
- **Analyste de tests (Bac +4/5)** : En SSII, l'analyste de tests devra s'adapter aux organisations dans lesquelles il est envoyé en mission. En interne dans une DSI, il sera une personne d'expérience. En plus de sa compétence dans le domaine fonctionnel, il est formé aux techniques de conception, aux outils de gestion, au processus de tests.
- **Analyste technique de tests (Bac +4/5)** : prend en charge les tests liés aux caractéristiques d'interopérabilité SI, de performance, de sécurité, l'automatisation des tests de non régression, l'analyse statique de code.
- **Gestionnaire d'environnements de tests** : Aguerri au POC (preuve de concept), il met en place l'outillage adéquat dans l'organisation. Certaines SSII se sont même spécialisées dans ce métier technique. Il prépare les scripts permettant de générer des rapports sur l'avancement des tests ou les anomalies...
- **Consultant tests et consultant senior** : indépendant ou en SSII. Il inter-

vient dans des projets au forfait ou en régie. Grande capacité à s'adapter aux situations. Il est amené à auditer les processus de l'entreprise, à proposer des axes d'amélioration, à prodiguer des formations, à répondre à des appels d'offres.

- **Chef de projet de tests** : c'est le gestionnaire des tests. Il organise les tests et l'équipe de testeurs. Il coordonne l'activité de son équipe, décide de l'automatisation de certains tests. Il suit les indicateurs et informe l'entreprise de l'avancement et de la qualité des systèmes en tests. Il participe aux comités directeurs des projets et y apporte la vision du test.
- **Responsable méthode et processus de tests** : son rôle est d'avoir une vision à long terme du test, des outils et des méthodologies. Il met en place la politique de tests de l'entreprise. Il assure la qualité logicielle et sa conformité.

Et le **développeur – testeur** ? Si sur le papier, l'approche peut être séduisante, en pratique, ce profil n'existe pas réellement et il est toujours préférable que le testeur soit indépendant du développeur même si ce dernier peut faire les tests de base comme les tests unitaires. Cependant, un développeur peut parfaitement devenir testeur.

Par exemple à l'ESIR (université de Rennes 1), les tests se retrouvent dans le cursus normal :

- 1^{re} année : approche informelle du test dans les travaux pratiques.
- 2^e année : formalisation du test, et mise en œuvre dans des stratégies de dévelop-

pement agile ou dirigées par le test.

- 3^e année : approfondissement avec une variante systèmes embarqués pour les élèves qui ont choisi cette orientation.

Allez-vous sauter le pas ?

François Tonic

Comprendre le **refactoring**

Modifier un code fonctionnel présente un risque. En effet il est possible d'y introduire des bogues et ainsi provoquer une régression de l'application. Le logiciel cesse alors de se comporter comme attendu.

Les bogues sont très souvent introduits parce que le code n'est pas lisible. On y trouve fréquemment des blocs d'instructions dupliqués à plusieurs endroits ou des structures conditionnelles complexes. Ces signes font partie de ce que l'on appelle *code smells*. Le compilateur ne se soucie pas de savoir si le code écrit est clair ou pas. Cependant toute modification du système implique un être humain, généralement un développeur, qui s'en préoccupe. Un système mal conçu est difficile à modifier, principalement parce qu'il n'est pas évident de savoir où le changement doit être effectué. S'il est difficile de savoir ce qu'il faut modifier, alors il y a de grandes chances que le programmeur fasse des erreurs et introduise des bogues. L'on peut être tenté de modifier le moins possible le système, après tout l'application est en production. Mais même si le programme marche, il complique la vie du développeur car ce dernier ne peut répondre convenablement aux besoins de son client. C'est pourquoi Abelson et al. [Abelson 1985] attirent notre attention sur le fait que les programmes doivent être écrits de sorte à être lus par les Hommes et non pas seulement pour être exécutés par les machines.

D'après Fowler et al. [Fowler 1999] la clé pour avoir un code lisible et facilement modifiable est le *refactoring*. Les auteurs de [Fowler 1999] remarquent que le refactoring fait de façon disciplinée augmente non seulement la confiance dans le code résultant, mais aussi permet au processus de se dérouler avec moins de stress. Notre objectif est d'introduire à l'aide de quelques exemples la notion de refactoring. Nous illustrerons comment améliorer la structure interne de l'application sans introduire de bogue dans le code. Cet article commence par définir la notion de *refactoring* pour ensuite introduire le concept de *code smell* sur lequel est basé tout refactoring. Finalement nous appliquerons le refactoring sur quelques extraits de code qui présentent des smells préalablement identifiés.

REFACTORING KÉSAKO?

Plusieurs définitions du terme refactoring émergent dans la littérature. Nous en retenons trois dans le cadre de cet article. Pour **Wake** [Wake 2004], le refactoring c'est l'art d'améliorer de façon sûre la conception du code existant. **Fowler** [Fowler 1999] pense que le refactoring c'est modifier le code source d'une application informatique pour améliorer sa structure interne de sorte que ce changement n'altère pas son comportement externe. Pour **Beck** [Fowler 1999], le refactoring est le processus qui consiste à prendre un programme fonctionnel et à lui apporter de la valeur, non pas en changeant son comportement mais en lui donnant les qualités qui permettent de continuer à développer rapidement. A la lecture de ces définitions nous tirons les enseignements suivants :

- Le refactoring ce n'est pas réécrire de zéro. C'est améliorer le code existant. Il n'est pas question de prendre le risque de recommencer from scratch.
- Le refactoring n'entraîne pas un ajout ou une suppression de fonctionnalités à un code. Le comportement de l'application avant et après le refactoring doit rester inchangé.

- Avec le refactoring, la conception de l'application se fait tout au long du développement. Ce qui n'empêche pas bien sûr une première phase de conception.

- Le refactoring permet de développer rapidement.

Le but du refactoring est d'avoir des programmes qui sont faciles à lire, dont les codes ne sont pas dupliqués, qui permettent des modifications sans mettre en danger les fonctionnalités existantes et qui permettent des logiques conditionnelles aussi simples que possible.

REFACTORING, OUI MAIS À QUELLES OCCASIONS ?

Une des questions qui vient à l'esprit quand il s'agit de faire du refactoring est celle de savoir le moment indiqué pour cette tâche. La pratique nous recommande de ne pas prévoir un créneau particulier pour le refactoring. En effet l'on fait du refactoring parce qu'on a besoin de rendre un service et le refactoring nous aide à réaliser ce besoin. Fowler et al. [Fowler 1999] identifient trois occasions idéales pour faire du refactoring.

- **Au moment de faire évoluer le système**

Nous l'avons déjà évoqué. Le refactoring ne change pas le comportement de l'application. Il n'y a donc pas ajout de fonctionnalités après un refactoring. Cependant, si la conception de l'application nous empêche d'y apporter une nouvelle fonctionnalité, alors refactorer le code pour faciliter cette évolution est recommandé.

- **Lors de la correction d'un bogue**

Corriger un bogue nécessite la compréhension du code existant. Si la conception de ce dernier ne facilite pas sa lisibilité, alors c'est le moment de le refactorer afin d'améliorer sa clarté. Le fait que notre programme présente une anomalie est un signe que notre code est peut être complexe, ce qui nous a empêché de voir le bogue.

- **Lors d'une revue de code**

La revue de code permet de partager les connaissances au sein de l'équipe de développement. Il donne la possibilité à d'autres personnes de faire des suggestions sur notre travail. C'est une pratique importante pour l'écriture de codes lisibles. En effet notre code peut nous paraître clair, mais pas à d'autres. Le reviewer procède alors au refactoring du programme afin de faciliter sa compréhension. Il est important que le reviewer et l'auteur du code y travaillent ensemble. Le reviewer fait des propositions de modifications et s'il est possible de les prendre en compte, alors la paire les met en place.

DETTE TECHNIQUE

Très souvent lorsque nous voulons corriger un bogue ou faire évoluer notre application deux situations se présentent à nous. La première implique une intervention rapide mais indigeste – nous sommes sûrs qu'elle va rendre difficiles les futures modifications. La seconde implique un code plus clair, mais demande plus de temps pour sa mise en place. La métaphore *dette technique* a été formulée par Ward Cunningham [Cunningham 1992] pour nous aider à

réfléchir sur ce problème. D'après cette métaphore, faire les choses rapidement et de façon indigeste nous apporte des dettes techniques. Comme toute dette celle-ci est aussi accompagnée des intérêts qui augmentent avec la durée de remboursement. Les coûts supplémentaires de maintenance et d'évolution du code résultant représentent les intérêts de cette créance. Le refactoring est alors le moyen de payer le capital. Afin de payer ce capital, il faut connaître le montant. Ce dernier est donné par l'identification des codes smells.

CODE SMELL

Un *code smell* ou *smell* est un indice superficiel qui correspond souvent à un problème plus ou moins profond dans le logiciel. Le terme a été pour la première fois avancé par Kent Beck dans [Fowler 1999]. L'indice est superficiel dans la mesure où il doit être rapide de l'identifier. Les commentaires qui décrivent ce que fait une méthode sont un exemple de smell. Leur présence est probablement due au manque de clarté du code de la méthode. Le smell n'implique pas toutefois toujours un problème. Par exemple les commentaires ne sont pas proscrits des programmes. Les utiliser en occurrence pour préciser pourquoi l'on prend une décision donnée fait partie des usages recommandés.

Examiner les smells les uns après les autres c'est un bon moyen d'apprendre aux équipes à mieux programmer. Dans la suite nous considérons des extraits de code qui comportent à chaque fois un smell. Notre but sera de corriger le smell en suivant le cycle de refactoring qui consiste pour chaque smell à :

- Ecrire, si ce n'est pas déjà fait, un ensemble de tests unitaires pour la section de code concernée. Tous les tests de cet ensemble doivent être passants. Il est essentiel pour le refactoring d'avoir de bons tests. En effet les tests donnent la sécurité et la confiance dont on a besoin pour modifier le programme.
- Dès que les tests sont effectifs, il faut sélectionner le refactoring qui va adresser le smell. Dans [Fowler 1999] les auteurs donnent un catalogue de smells et à chaque fois la suite de refactorings à appliquer pour les effacer.
- Appliquer le refactoring sélectionné. Chaque cycle de refactoring se termine par l'exécution des tests unitaires. Si tous les tests passent alors nous avons apporté de la qualité au produit sans introduire de bogue, nous avons fait du refactoring.

EXEMPLES DE REFACTORING

Nous allons fournir dans cette section quelques exemples de refactoring. Tous nos exemples sont donnés en java.

> Exemple 1 : Encapsulation des attributs

Le but de l'encapsulation est de contraindre les clients d'un objet à passer par des accesseurs pour utiliser ses attributs plutôt que de les utiliser directement. Considérons la classe *Personne* suivante :

```
public class personne { public String nom ;}.
```

Et la méthode de tests donnée comme suit :

```
public void testPersonne(){
    Personne personne = new Personne();
    personne.nom = «Majirus Fansi»;
    assertEquals(«Majirus Fansi», personne.nom);
}
```

Nous constatons que la propriété nom de la classe *Personne* n'est pas encapsulée (modificateur *public*). Il s'agit d'un smell. Le catalogue de Martin Fowler nous conseille de procéder de la façon suivante pour lever ce smell.

- Créer les accesseurs en lecture (*getter*) et écriture (*setter*) pour ce champ.
- Localiser toutes les références à ce champ et remplacer toutes les lectures par les appels à l'accesseur en lecture et toutes les écritures par les appels à l'accesseur en écriture.
- Compiler et tester après chaque changement de référence.
- Déclarer le champ comme privé (modificateur *private*).
- Compiler et tester.

Ce refactoring comme tout refactoring se fait donc par successions de petites étapes. C'est le fait de procéder par petites étapes successives qui nous donne la confiance dans les résultats.

Etape 1 : création du getter et du setter (voir listing 1).

```
public class Personne {
    private String nom;
    public String getNom () {
        return nom;
    }
    public void setNom (String unNom){
        nom = unNom;
    }
}
```

Listing 1: création des accesseurs

Etape 2 : Rechercher les références directes au champ nom et les remplacer par les accesseurs. Une façon d'y parvenir serait de rendre temporairement ce champ privé (modificateur *private*). Toutes les références au champ seront alors en erreur. Nous veillons à remettre ce champ public avant toute modification. Notre environnement de développement favori peut aussi nous aider à trouver toutes les références d'une propriété.

Remplacer l'affectation *personne.nom = «Majirus Fansi»* par *personne.setNom(«Majirus Fansi»)*.

Etape 3 : compiler et tester.

Etape 2 : remplacer l'accès *personne.nom* par *personne.getNom()*.

Etape 3 : compiler et tester.

Etape 4 : une fois tous les clients modifiés, rendre le champ nom privé (voir listing 2).

```
public class Personne {
    private String nom;
    public String getNom () {
        return nom;
    }
    public void setNom (String unNom){
        nom = unNom;
    }
}
```

Listing 2: encapsulation

Etape 5 : compiler et tester une dernière fois. Le test est toujours passant et le champ nom est maintenant encapsulé.

> Exemple 2 : Méthode longue

Une méthode est considérée longue quand elle comporte plus d'une douzaine de lignes de code. Fowler [Fowler 1999] conseille d'utiliser le refactoring nommé *Extract Method* pour la simplifier en petites méthodes. Le but n'est pas d'extraire à chaque fois un certain nombre de lignes mais de créer à chaque fois une nouvelle méthode avec un bloc d'instructions ayant un sens sémantique. Il faut garder à l'esprit que le fait que la méthode soit longue est juste un smell. En effet il se pourrait que ce soit la meilleure façon d'exprimer une idée. Considérons la classe *Report.java* donnée en listing 3. La méthode *report* de cette classe compte plus de douze lignes. Elle est donc

longue, sa compréhension en est compromise. Nous nous proposons de la simplifier en appliquant le refactoring *extract method*. La méthode *report* fait apparaître deux blocs d'instructions. Le premier s'intéresse au rapport sur la liste des machines passées en argument. Le second s'occupe du rapport d'un objet robot. Pour des raisons de limitation d'espace nous ne définissons pas ici les classes *Machine* et *Robot*. Nous supposons aussi que la classe de tests qui assure la non régression de la méthode *report* est passante.

```
public class Report {
    public static void report(Writer out, List<Machine> machines, Robot robot) throws
    IOException {

        out.write("FACTORY REPORT\n");

        /** debut report machine**/
        Iterator<Machine> line = machines.iterator();
        while (line.hasNext()) {
            Machine machine = (Machine) line.next();
            out.write("Machine " + machine.name());

            if (machine.bin() != null)
                out.write(" bin=" + machine.bin());

            out.write("\n");
        }
        out.write("\n");
        /** fin report machine**/
        /** debut report robot**/
        out.write("Robot");
        if (robot.location() != null)
            out.write(" location=" + robot.location().name());

        if (robot.bin() != null)
            out.write(" bin=" + robot.bin());

        out.write("\n");
        /** fin report robot**/
        out.write("=====\n");
    }
}
```

Listing 3 :

Nous créons une première méthode (*reportMachines*) pour extraire le traitement lié aux machines. Cette méthode est donnée en listing 4.

```
private static void reportMachines(Writer out, List<Machine> machines) throws IOException {
    Iterator<Machine> line = machines.iterator();
    while (line.hasNext()) {
        Machine machine = (Machine) line.next();
        out.write("Machine " + machine.name());
        if (machine.bin() != null)
            out.write(" bin=" + machine.bin());

        out.write("\n");
    }
    out.write("\n");
}
```

Listing 4: extract method *reportMachines*

Après compilation et test, nous pouvons procéder à l'extraction du second bloc. Celui qui traite du robot (*reportRobot*). Cette dernière est donnée en listing 5.

```
private static void reportRobot(Writer out, Robot robot) throws IOException {
    out.write("Robot");
    if (robot.location() != null)
        out.write(" location=" + robot.location().name());

    if (robot.bin() != null)
        out.write(" bin=" + robot.bin());

    out.write("\n");
}
```

Listing 5 : extract method *reportRobot*

Nous compilons et lançons à nouveau notre classe de tests. La méthode *report* résultant de ce refactoring est donnée en listing 6. Après ce refactoring la méthode *report* compte quatre lignes de

```
public static void report(Writer out, List<Machine> machines, Robot robot) throws
    IOException
{
    out.write("FACTORY REPORT\n");
    reportMachines(out, machines);
    reportRobot(out, robot);
    out.write("=====\n");
}
```

Listing 6 : méthode *report* après refactoring

code. Nous remarquons que sa lisibilité est améliorée. Le résultat de cette première étape nous montre que nous pouvons aller encore plus loin dans la conception de notre programme en créant par exemple une méthode *report()* dans les classes *Machine* et *Robot*. Chacune de ces classes aura alors la responsabilité d'afficher son rapport. Nous avons fait le choix de présenter ces exemples de refactoring indépendamment de tout environnement de développement intégré (IDE – Integrated Development Environment). Sachez que votre IDE peut vous aider à effectuer rapidement ces tâches. Bien sûr il faut savoir, pour bien profiter des facilités apportées par l'environnement de développement, quel refactoring vous souhaitez appliquer, pourquoi vous désirez le faire et quel résultat vous visez. Le catalogue de refactorings est donné dans [Fowler 1999]. Vous y trouverez des refactorings qui ne sont pas fournis par votre IDE. La plupart de ceux-ci vous permettent de simplifier la conception de votre programme en appliquant les patterns de conception [Gamma 1994]. Wake fournit dans [Wake 2004] des exercices pour se familiariser à cette technique.

CONCLUSION

Programmer est à plusieurs égards une conversation avec un ordinateur. Nous écrivons le code qui donne des instructions à l'ordinateur et il répond en faisant exactement ce que nous lui avons demandé de faire. Par moment nous réduisons l'écart entre ce que nous souhaitons qu'il fasse et ce que nous lui demandons de faire. Programmer c'est exprimer ce que l'on veut. Mais il existe un autre utilisateur de notre code source. Quelqu'un qui va lire notre programme dans quelques mois pour essayer de le faire évoluer. Nous oublions souvent cet autre utilisateur, qui se révèle être le plus important pour la survie de notre logiciel. En effet personne ne se soucie de savoir si l'ordinateur a besoin de quelques cycles supplémentaires pour compiler un code, mais se préoccupe de savoir que le développeur consacre une semaine à une modification qui aurait nécessité une heure s'il avait compris notre code. Le problème est que lorsque nous essayons de faire marcher notre programme, nous ne pensons pas à ce futur développeur. Il est impératif de changer notre façon de développer pour rendre notre code aisément compréhensible. Le refactoring nous aide à rendre notre programme plus lisible. Dans cet article, nous avons expliqué ce qu'est le refactoring. A travers quelques exemples simples et accessibles nous avons illustré le processus de refactoring. Ceux que nous avons choisis étaient en Java, mais le refactoring s'applique quel que soit notre langage de programmation.



Majirus Fansi, PhD

Consultant SOA/Java EE à Valtech technology Paris

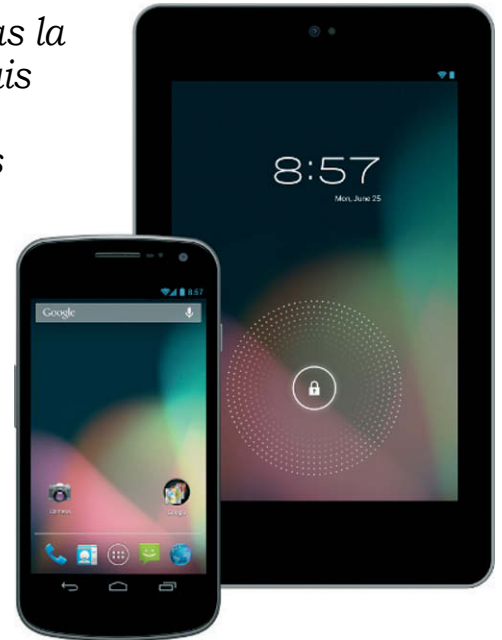
Références :

- [Abelson 1985] H. Abelson, G. Sussman and J. Sussman **The Structure and Interpretation of Computer Programs**, 1985.
- [Cunningham 1992] Cunningham Ward. **The WyCash Portfolio Management System**, OOPSLA'92 Experience Report.
- [Fowler 1999] Fowler M., Beck K., Brant J., Opdyke W., Roberts D. **Refactoring, Improving the design of existing code**, Addison Wesley, 1999.
- [Gamma 1994] Gamma E., Helm R., Johnson R., Vlissides J. **Design Patterns, Elements of Reusable Object-Oriented Software**, Addison Wesley, 1994.
- [Wake 2004] Wake William C. **Refactoring Workbook**, Addison Wesley, 2004.

Android 4.1 : le fonctionnement du Project Butter

La nouvelle version d'Android vient de sortir ! Ce n'est pas la version 5, comme annoncé un peu partout sur le web, mais bien une 4.1 appelée Jelly Bean (en attendant "Kouign-amman" ?). Si elle apporte un grand nombre de nouvelles fonctionnalités telles que Google Now, un système de notifications étendues, la recherche vocale etc., celle-ci nous arrive surtout avec des promesses de performances et fluidité accrues qui feraient presque pâlir un Galaxy SIII à côté d'un vieux Nexus S sous Jelly Bean.

Ces améliorations sont dues à un projet que les ingénieurs de Google ont surnommé le "Project Butter" et qui peut se résumer à un ensemble d'optimisations logicielles portant l'expérience Android à un tout autre niveau de qualité.



Nous allons présenter dans cet article, les grands principes des améliorations apportées dans le cadre de ce projet. Cependant, le Project Butter a ceci de particulier qu'il n'est pas à proprement parler utilisable directement par les développeurs, on ne pourra donc pas voir **comment** rendre notre application plus fluide, mais **comment Google a fait** pour que les applications qui tournent sous Android soient plus fluides intrinsèquement.

Par conséquent, il n'y aura pas de code présenté ici, mais plutôt une explication des concepts de ces optimisations.

Signalons aussi que cet article est grandement inspiré de la conférence tenue par Romain Guy et Chet Haase au Google I/O 2012, et que la plupart des illustrations en sont même tirées.

Bonne lecture.

OPTIMISATIONS SYSTÈME

Il existe plusieurs façons d'améliorer la sensation de fluidité d'un système d'affichage. Dans cette nouvelle version d'Android, les ingé-

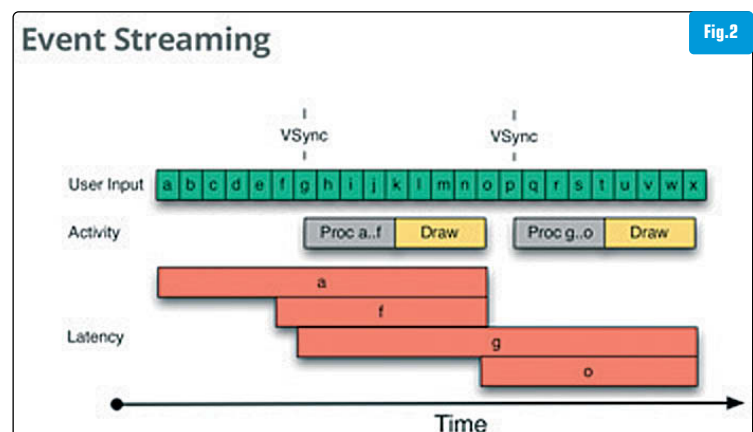
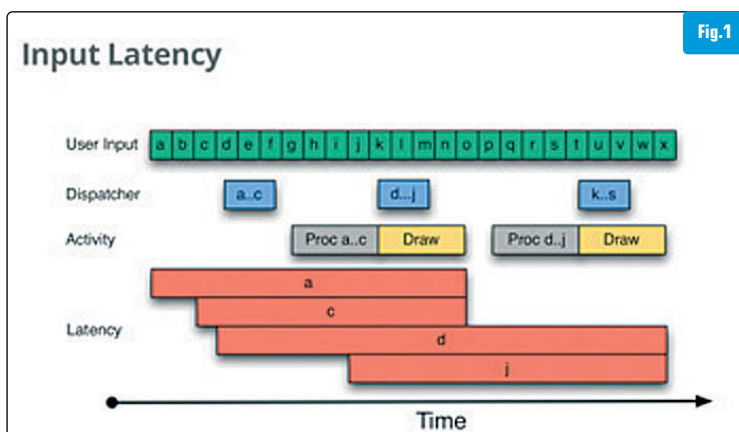
nieurs de Google en ont choisi au moins 3 :

- réduire la latence d'un évènement
- maintenir l'affichage des frames synchronisé avec le traitement de celles-ci
- augmenter la vitesse de rendu des vues.

Voici quelques explications plus précises.

> Réduction de latence

Pour ceux qui ne le savent pas, la latence se définit par le temps passé entre le moment où quelque chose se passe (l'utilisateur touche l'écran par exemple), et le résultat de cette action diffusée à l'écran. Dans les précédentes versions, Android empilait les évènements en entrée, les envoyait par groupe (ou batch) dans une file d'évènements pour les dessiner ensuite à l'écran. C'est ce qui est représenté [Fig.1] : une série d'évènements (a..x) est déclenchée par l'utilisateur; ces évènements sont groupés puis passés au Dispatcher qui va les envoyer à un thread au niveau de l'activité pour les



dessiner. On peut facilement observer le problème de latence sur cette figure : l'utilisateur ne voit les conséquences de l'évènement **a** (de même pour **c** puisqu'ils sont dans le même batch) qu'à la fin du premier "Draw".

Pour résoudre ce problème, il a fallu remplacer certaines couches en charge du groupement ou du dispatch par un système de streaming pour essayer d'envoyer les évènements le plus rapidement possible au système de rendu. C'est ce qui est démontré [Fig.2]. Au moment du process (Proc a..f), le système récupère tous les évènements antérieurs (et non plus seulement le premier batch [a..c] comme précédemment) et les dessine.

La latence est toujours importante pour les plus anciens évènements (**a**), mais largement réduite pour les plus récents (**f**). La latence moyenne générale est donc bien améliorée.

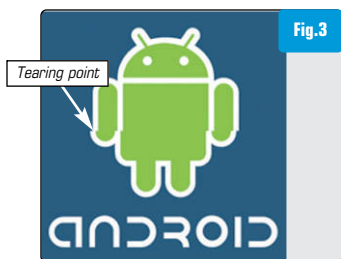
Comme cela n'était pas suffisant, Google a aussi ajouté un système de prédiction d'évènements permettant d'anticiper, dans certains cas (comme lors d'un scroll de listview par exemple), le prochain évènement d'entrée.

> VSync

Pour expliquer ce qu'est VSync, un léger rappel sur le fonctionnement de l'affichage s'impose : une vidéo (ou une animation), est composée d'images individuelles, appelées frames, affichées les unes à la suite des autres.

Les frames sont faites de pixels, et lorsque l'écran affiche une frame, les pixels sont remplis ligne par ligne. L'écran reçoit chaque frame de la puce graphique et commence donc à la dessiner ligne par ligne.

On souhaite par conséquent que lorsque l'écran termine de dessiner une frame, il récupère la suivante depuis la puce graphique, l'affiche, et ainsi de suite.... Le phénomène appelé "tearing" (cf [Fig.3]), survient lorsque la puce graphique envoie une nouvelle frame à l'écran au milieu du processus de dessin d'une autre frame. Dans ce cas, on voit apparaître à l'écran une partie de la première frame et une partie de la suivante [Fig.3].



VSyn est normalement utilisé pour prévenir le GPU qu'il doit attendre la fin du processus de dessin ligne par ligne de l'écran avant d'envoyer la frame suivante. Il ne sert en fait qu'à synchroniser l'envoi de frame par la puce graphique et le processus de dessin de celle-ci, prévenant l'effet de tearing.

VSyn était déjà présent dans les précédentes versions d'Android, mais il n'était utilisé, jusqu'à maintenant, que pour maintenir synchronisé l'écran avec une petite partie bas niveau du framework (le SurfaceFlinger qui est en charge de "récupérer" toutes les fenêtres qui vont être affichées à l'écran et d'en faire une image). Ceci était suffisant pour prévenir le tearing.

On considère en général qu'une animation est fluide lorsqu'elle est affichée autour de 60 frames par secondes. Cela fait une frame à calculer et à afficher toutes les 16ms. C'est peu, et si le système prend plus que 16ms, il affichera 2 fois la même image (pendant 32 ms) et perdra un peu de fluidité. Dans la mise à jour précédente (Ice cream sandwich), le début du processus de rendu d'une image démarrait plus ou moins aléatoirement quand la précédente avait terminé (cf. [Fig.4]).

Dans l'image [Fig.4], on peut voir un exemple de rendu sans VSync. L'image 0 est affichée à l'écran pendant que le CPU, puis le GPU calculent l'image 1. Au rafraîchissement de l'écran (après 16 ms), l'image 1 est prête et est donc affichée, parfait. Puis le calcul de l'image 2 commence mais prend trop de temps (ou commence trop tard, selon l'interprétation) pour être affichée à l'écran lors de l'impulsion suivante.

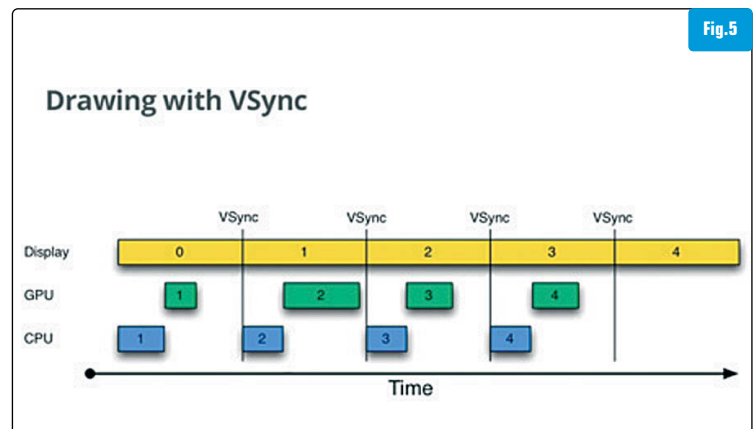
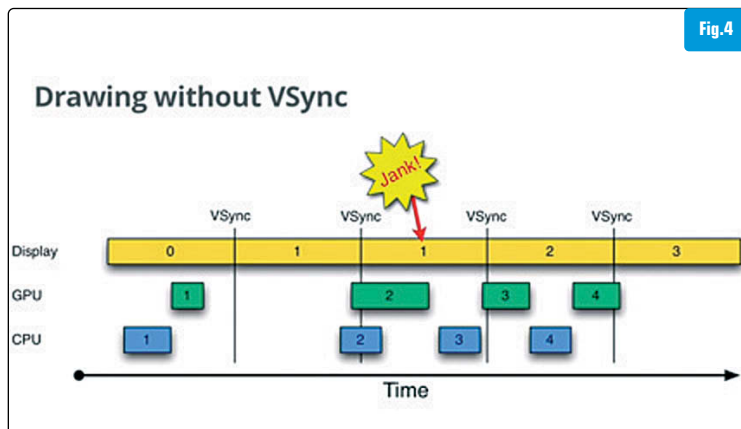
C'est par conséquent toujours l'image 1 qui est forcée d'être affichée durant de nouveau 16ms. Il y a donc un "lag" de l'animation. Une conséquence perverse de ce problème est que peu importe la rapidité de votre code de dessin, si vous dessinez une vue au moment de l'impulsion VSync, vous allez voir la précédente 2 fois à l'écran. Et même en utilisant alors tous les outils de Debug à disposition, vous n'auriez détecté aucun problème à votre code et n'auriez rien pu faire pour améliorer la fluidité.

Google a réparé cela pour vous, puisque les ingénieurs ont décidé qu'il fallait étendre l'utilisation de VSync au framework entier (toutes les fonctions de dessin, les animations, le scroll etc.) afin de démarer le rendu d'une image dès que celui de la précédente est fini, au moment de l'impulsion VSync (cf. [Fig.5]).

Dans l'image [Fig.5], on peut voir la même situation que pour la fig.4, mais utilisant cette fois VSync. Le calcul de chaque image commence à chaque impulsion de VSync, toutes les 16ms. Les calculs prennent moins de 16ms, tout se passe bien dans le meilleur des mondes, et les animations sont "butter smoothy".

Donc si avec Jelly Bean, une de vos animations devient lente, vous pouvez à présent utiliser les outils de Debug pour voir pourquoi et comment optimiser votre code, puisque dorénavant, c'est probablement votre faute.

Ce système permet donc de laisser le maximum de temps possible



(16ms) pour calculer une image. Mais que se passe-t-il si le calcul d'une image prend plus de 16ms ? Les animations sont saccadées de nouveau. Il existe pour cela une technique, elle aussi ajoutée dans JB, appelée triple buffering.

> Triple Buffering

Pour comprendre ce qu'est le triple buffering, deux notions sont à maîtriser : ce qu'est un buffer, et ce qu'est le double buffering.

Buffer :

Jusqu'à maintenant, on a parlé de frames qui sont calculées puis affichées à l'écran, c'est vrai mais on peut être plus précis. Ce sont en fait les informations de ces frames qui sont contenues dans des buffers qui sont affichées à l'écran. Un buffer, schématiquement, n'est qu'un conteneur dans lequel les informations relatives aux frames sont construites et stockées. Ces buffers sont ensuite "dépiplés", pour afficher ces informations à l'écran.

Double buffering :

C'est une technique classique, qui consiste à employer 2 buffers et qui veut simplement dire qu'un écran peut afficher une frame d'un buffer, pendant que le reste du système travaille à construire la frame suivante dans l'autre buffer. Une fois prêtes, les informations des buffers sont échangées (swap) : lors de l'impulsion VSync, le 2^e buffer est affiché et le premier nettoyé et prêt à être utilisé pour travailler de nouveau (cf. [Fig.6]).

C'est un peu comme si vous vouliez remplir une piscine (l'écran) avec de l'eau (des informations) et un robinet qui coule en permanence (le flux d'informations) avec 2 seaux (les buffers) à intervalles réguliers (l'impulsion VSync). Il vous suffit de remplir un seau pendant que vous allez vider le premier dans la piscine, puis, de les échanger, et ainsi de suite. Cela vous permet de remplir la piscine sans arrêter le flux d'informations. C'est un processus très employé qui fonctionne correctement quand les applications se "comportent" bien.

Triple buffering :

Si tout ne se passe pas comme prévu, qu'un buffer n'est pas prêt pour être swappé au moment de l'impulsion VSync (il prend plus que 16 ms pour être prêt), la même information est affichée durant 2 périodes VSync à l'écran (cf. [Fig.7]).

Et on voit bien sur cette même figure qu'il existe un laps de temps durant lequel le CPU ou le GPU sont inactifs. Il faut donc pouvoir tirer parti du temps "libre" de certains composants, en commençant le travail à venir, pour que les buffers soient prêts, non pas plus rapidement, mais plus en avant dans le temps.

C'est la technique dite du triple buffering. Elle consiste à ajouter un

buffer et... c'est tout. Le principe est le même que pour du double buffering, mais avec un buffer en plus, qui permet aux composants libres de commencer à travailler sur ce buffer.

Si on reprend la métaphore du seau mais qu'on donne maintenant 3 seaux et qu'on identifie les CPU et GPU par 2 personnes Cédric (CPU) et Ghislain (GPU).

Cédric s'occupe de remplir les seaux en premier, les passe à Ghislain, qui va remplir la piscine au moment de l'impulsion. Si Ghislain ne revient pas alors que le seau B est presque prêt, Cédric peut quand même commencer à remplir le seau C etc... Et la piscine aura toujours un seau de prêt à côté d'elle au moment de l'impulsion. Il peut en résulter un lag lorsque le système rentre dans cet état, mais les performances seront bien meilleures après et cela permet au moins de stopper les redondances de lags toutes les 2 périodes comme avec le double buffering.

Pourquoi Jelly Bean n'utilise pas cette technique constamment ? Comme on peut le voir sur la [Fig.7], et comme nous venons de le dire, le triple buffering introduit un petit lag dans le processus de rendu. Donc lorsque les choses ne vont plus très bien, on a le choix entre 2 mauvaises options : un petit lag au début d'un événement (par exemple le clic-long sera un tout petit plus long), ou une animation saccadée.

C'est pourquoi Jelly Bean n'utilise la technique du triple buffering que lorsque c'est réellement nécessaire, il tourne normalement avec un double buffer et, lorsque les choses se passent mal, on peut utiliser le 3^e. Il y a donc moins de lag d'entrée et si besoin, le 3^e buffer aide le système à "récupérer".

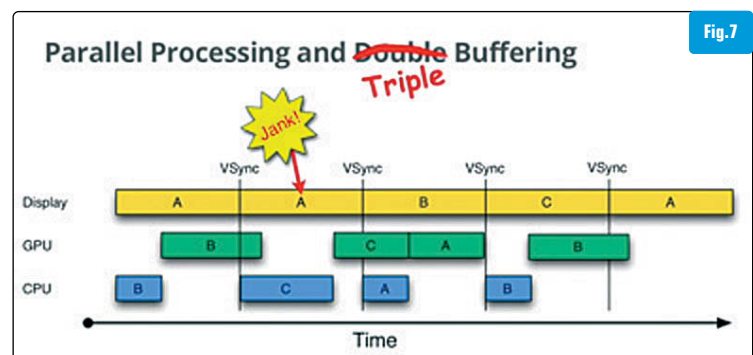
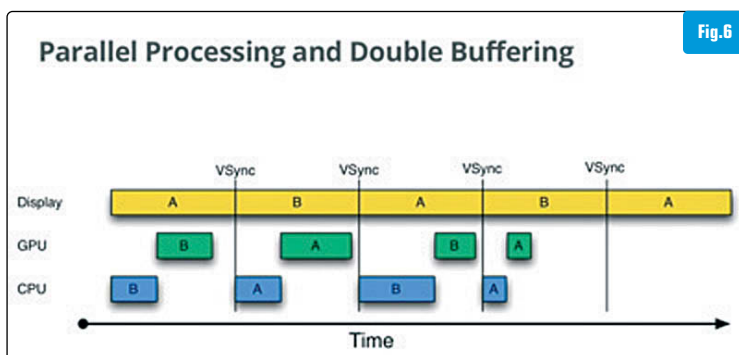
CONCLUSION

Comme nous venons de le voir, Google continue d'améliorer son système mobile au fil des versions d'Android. Cette version arrive avec des modifications dont la plupart sont des améliorations significatives en termes de performances même si il est très difficile de rendre compte à quel point elles changent toute l'expérience utilisateur. Le mieux est encore de prendre en main un appareil sous ICS et un autre sous JB, d'y effectuer des actions similaires et de comparer les impressions.

Même si nous ne pouvons pas utiliser directement dans notre code ces petites améliorations et qu'il est évident que le survol de quelques-unes de ces modifications n'est pas suffisant pour comprendre en profondeur toutes les optimisations apportées, cet article avait pour but que vous les appréhendez un peu mieux et surtout de vous donner envie de les regarder de plus près.

Paul Marois

Ingénieur Conception et Développement chez Genymobile.



Petit aperçu du développement SharePoint 2013

En juillet, Microsoft a levé le voile sur la nouvelle mouture de SharePoint et l'a mise à disposition en téléchargement, version "Preview". Si les fondations n'ont finalement pas vraiment changé, certains domaines ont été clairement creusés. Cet article vous présentera les principales nouveautés pour les développeurs SharePoint.

UN OUTILLAGE RÉACTUALISÉ

> Une création de projet simplifiée avec Visual Studio 2012

Evidemment, un des premiers gros changements concerne l'environnement de développement, SharePoint 2013 nécessitant Visual Studio 2012. Ce dernier est disponible en version finale depuis le 15 Août dernier, vous pouvez donc d'ores et déjà l'installer et l'utiliser pour vos développements. Sachez cependant qu'il vous faudra télécharger les "Microsoft Office Developer Tools for Visual Studio 2012" (toujours en Preview) à l'adresse suivante : <http://bit.ly/O2013Tools>, afin de disposer des modèles de projets.

Microsoft a retiré une partie des modèles de projets pour se concentrer principalement sur un projet principal, si on exclut ceux pour Silverlight, des WebParts visuels ou les imports de package ou de workflows SharePoint Designer. Le modèle "SharePoint 2013 Project" reste très versatile puisqu'on pourra y ajouter ensuite les éléments désirés : Web Parts, définitions de site, modèles de listes... On notera au passage que ces projets utilisent désormais le Framework .NET 4.5 [Fig.1 et 2].

Microsoft a continué à travailler sur les Designers des éléments de projet. On trouve ainsi des éditeurs pour les modèles de liste et les types de contenu dans lesquels on peut personnaliser facilement les colonnes, affichages et types de contenus associés. Pour le reste, vous ne serez pas dépayés, mis à part le nouveau look de Visual Studio 2012 bien entendu [Fig.3 et 4].

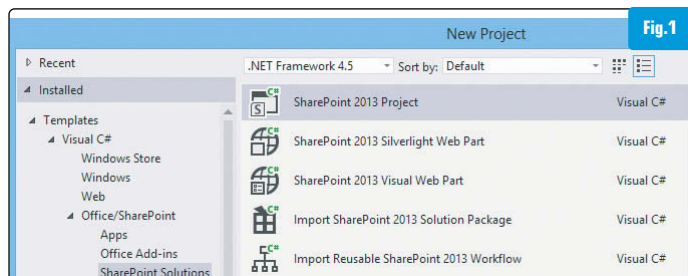


Fig.1

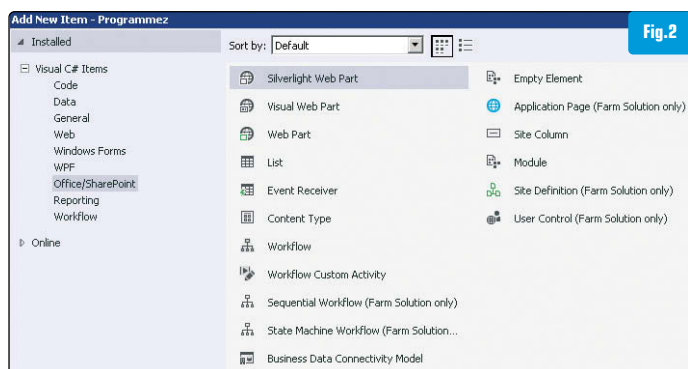


Fig.2

> Et SharePoint Designer ?

Quelques petits mots sur cet outil de conception et de personnalisation de SharePoint qui, s'il n'est pas dédié aux développeurs, demeure un bon complément pour ceux-ci.

Dans sa nouvelle version, le plus gros changement visible vient du retrait du designer Web (la vue "code" perdue quant à elle pour les utilisateurs avertis). En effet, une des utilisations de SharePoint Designer concernait l'intégration graphique puisqu'il était possible de modifier les pages, pages maîtres et mises en pages avec un rendu en live. Si cela ne va pas manquer de faire grincer quelques dents, cela va de pair avec une nouveauté de SharePoint 2013 : le Design Manager. Grâce à lui, il est possible de travailler sur l'aspect graphique de son site avec ses applications habituelles (Dreamweaver,

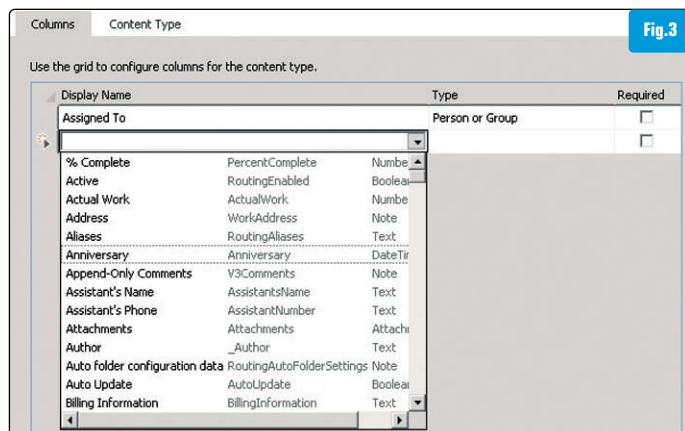


Fig.3

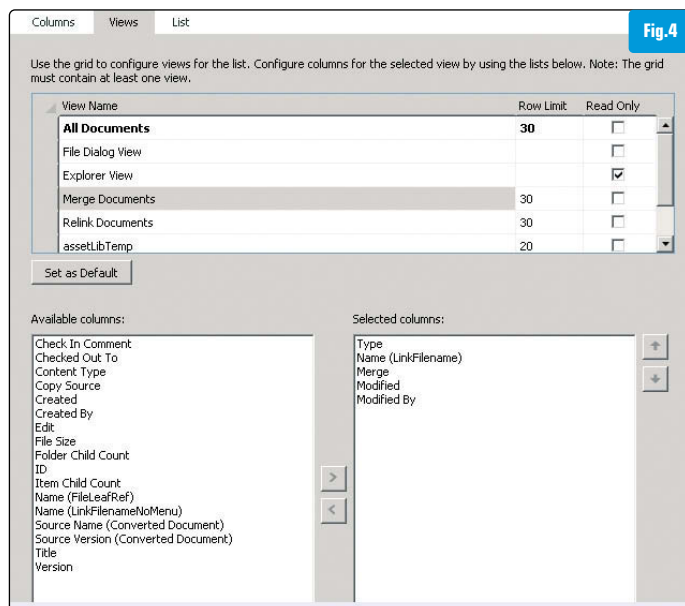


Fig.4

Expression Web, ...) via un partage réseau vers SharePoint et son seul navigateur web, SharePoint s'occupant de transformer à la volée la page HTML en page maître ou en mise en page. Des snippets sont même disponibles pour intégrer des composants .NET ou SharePoint [Fig.5].

A part cela, c'est principalement du côté des workflows que les choses ont évolué. Ceux-ci tirent parti de l'infrastructure fournie par Windows Azure Workflow et les Frameworks .NET 4 et 4.5. Ils autorisent maintenant les modèles "machine à état", plus souples et plus adaptés à la gestion documentaire que les traditionnels workflows séquentiels. D'autres ajouts comme les boucles (while, for), l'appel de services web et REST, la gestion de dictionnaires de variables ou bien encore un packaging amélioré apportent un vernis appréciable. Enfin, on notera que Visio et SharePoint Designer se sont encore rapprochés puisqu'il sera possible de bénéficier du "Visual Workflow Designer" au sein de SharePoint Designer qui n'est autre qu'un Active X Visio offrant une expérience plus riche et pratique.

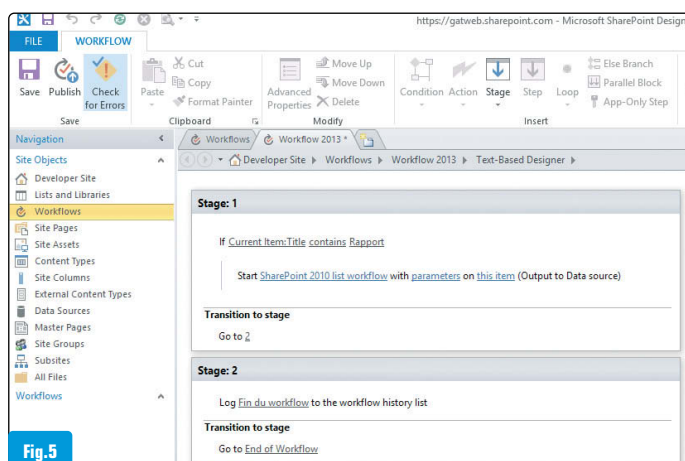
> Un modèle objet client étendu

Un des chantiers amorcés avec SharePoint 2010 était la mise en place d'un modèle objet client permettant de consommer et manipuler sites et listes SharePoint depuis des applications clientes en Javascript ou code managé .NET. L'adoption de ce nouveau type de développement est allée crescendo car cela évite bien souvent d'avoir à déployer quoi que ce soit sur les serveurs, allant de pair avec l'utilisation plus fréquente de frameworks Javascript/AJAX comme jQuery ou Prototype.

Si cette API était limitée aux principales opérations (création de listes ou d'éléments, récupération de données, ...), il n'en est plus de même avec la version 2013. En effet, s'ajoutent à cette base la recherche d'entreprise, les réseaux sociaux et profils utilisateurs, la brique ECM (Enterprise Content Management, qui englobe à la fois la gestion documentaire avancée, la taxonomie, mais aussi la publication Web) ou encore les Business Connectivity Services. On retrouve comme avant la version Javascript et les versions .NET (classique, Silverlight mais aussi maintenant Windows Phone avec le support de notifications /pushs), ainsi qu'un petit nouveau (pour SharePoint) : REST.

Les API REST

SharePoint offre désormais un service REST (Representational State Transfert) qui s'apparente au modèle client. L'intérêt majeur



de ce service réside dans le fait qu'il s'affranchit des technologies puisqu'il s'appuie sur de simples requêtes HTTP pour pouvoir effectuer des opérations "CRUD" (Create Read Update Delete). On utilise ainsi des méthodes GET pour la lecture, POST pour les mises à jour ou DELETE pour les suppressions. On peut dès lors consommer des données issues de SharePoint depuis des programmes en .NET, Java ou PHP, sans difficulté.

Pour cibler l'objet sur lequel porte l'opération, il vous faudra construire une URL spécifique comme suit : [Fig.6].

Si la première partie reste similaire aux appels de Web Services SharePoint ou à son modèle objet client, on note ensuite un chemin propre à cette API : un répertoire virtuel "_api" et un chemin représentant une commande ou un objet. Ce dernier peut être "site", "web" ou encore "lists", suivi de la méthode et/ou des paramètres ou filtres supplémentaires. Voici quelques cas simples pour illustrer ceci.

Imaginons que vous souhaitez récupérer des informations depuis le site <http://sharepoint/programmez>.

Pour récupérer le titre du site, effectuez une requête en GET sur http://sharepoint/programmez/_api/web/title (vous pouvez aussi saisir directement l'URL dans votre navigateur puisque c'est la méthode par défaut).

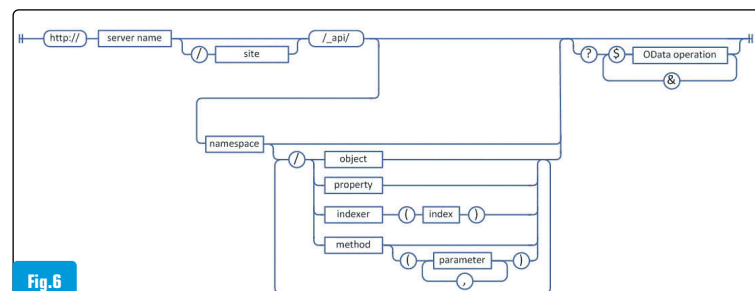
Vous voudriez plutôt connaître le propriétaire de la collection de sites ou son utilisation ? Cela correspondra respectivement aux URL http://sharepoint/programmez/_api/site/owner et http://sharepoint/programmez/_api/site/usage. Pour créer une nouvelle bibliothèque de documents, rien de plus simple, effectuez une requête POST sur http://sharepoint/programmez/_api/web/lists comme suit :

```
WebClient c = new WebClient();
c.UseDefaultCredentials = true;
string url = "http://sharepoint/programmez/_api/lists";
string content = @"{
  'metadata':{'type':SP.List},
  'BaseTemplate': 101,
  'Description': 'Tous mes documents',
  'Title': 'Mes docs'
}";
byte[] data = UTF8Encoding.UTF8.GetBytes(content);
Stream s = c.OpenWrite(url, "POST");
s.Write(data, 0, data.Length);
s.Close();
```

En connaissant le modèle objet SharePoint, il est très facile de comprendre et manipuler cette API, quelle que soit votre application.

> Les Apps

J'ai gardé le meilleur pour la fin, la nouveauté majeure pour les développeurs : les Apps pour SharePoint, et l'Office Store associé, le futur Marketplace pour Office et SharePoint. Ce magasin en ligne est la vitrine publique des Apps, mais sachez qu'il est aussi tout à fait possible d'avoir son propre catalogue d'entreprise dans lequel vous pourrez publier vos applications et autoriser - ou non - celles du Store officiel [Fig.7].



> Mais pourquoi des Apps ?

SharePoint est depuis longtemps connu pour son extensibilité et sa capacité à servir de socle pour de nombreuses applications internes à l'entreprise. Seulement, cela engendre souvent des problématiques de déploiement sur les serveurs et donc d'isolation (ou non) de ce code vis-à-vis des autres sites présents dans la ferme SharePoint. Une première réponse avait été donnée dans SharePoint 2010 grâce au système des solutions utilisateurs (ou "sandbox"). Les gestionnaires d'espaces SharePoint devenaient alors capables de gérer leurs propres packages sans passer par un circuit généralement plus long de déploiement global. Un service Windows dédié s'occupait de surveiller l'exécution du code afin d'éviter d'impacter les instances IIS en cas de souci et pouvait utiliser des quotas de ressources. Surtout, des contraintes fortes au niveau de la sécurité bloquaient toutes requêtes externes (web service, base de données) ainsi qu'une large partie de l'API, limitant par là même la consommation de ressources et les risques d'instabilité pouvant porter atteinte aux autres instances.

Mais ce système était trop cloisonné et trop contraignant pour pouvoir être utilisé à grande échelle. De plus, il lui manquait le cycle de vie d'une application : publication, mises à jour, retrait, catalogue, voire monétisation. Enfin, on restait forcément dans le contexte purement SharePoint, avec une intégration compliquée de services extérieurs. C'est de ce constat qu'est parti Microsoft pour concevoir les Apps.

> Qu'est-ce qu'une App ?

Une App est une application intégrée à SharePoint, pouvant communiquer facilement avec celui-ci, enrichir ses fonctionnalités, mais aussi dialoguer avec le monde extérieur. Elle se base sur des standards que sont HTML, CSS, Javascript et OAuth, ce dernier étant déjà largement utilisé dans l'industrie du Web pour l'extensibilité de services en ligne tels que Gmail, Facebook, LinkedIn ou Twitter. OAuth répond aussi au besoin de pouvoir agir en tant qu'application avec les droits qu'on lui a concédé, et non en tant qu'utilisateur.

L'architecture des Apps repose sur un système ouvert et en tant que développeur ou éditeur de logiciel, vous aurez le libre choix du mode d'hébergement : uniquement dans SharePoint (SharePoint-

hosted), au sein d'une instance Azure associée à SharePoint (Auto-hosted) ou bien celui de votre préférence (Provider-hosted) [Fig.8].

Le mode "SharePoint-hosted" est relativement classique puisqu'il se rapproche le plus de ce que l'on avait l'habitude de faire avec la version 2010, la principale différence étant que l'App crée automatiquement un sous-site dédié, isolant son contenu du reste de la collection.

L'"Autohosted" profite des services Windows Azure qui peuvent être associés à SharePoint pour disposer d'une infrastructure à la demande. Il sera ainsi possible de provisionner des sites et des bases SQL Azure et de les exploiter conjointement à SharePoint. Il est d'ailleurs prévu qu'Office 365 propose ce service de manière totalement intégrée.

Enfin, le mode "Provider-hosted" sera sans doute le préféré de ceux qui possèdent déjà leur infrastructure, puisque vous pourrez intégrer directement vos services et vos pages, qu'ils soient écrits en PHP, Java ou .NET, pourvu qu'ils supportent OAuth.

> La composition d'une App

Une App commence comme toujours par un projet sous Visual Studio : [Fig.9]. A l'intérieur de ce projet, vous trouverez un fichier très important : "AppManifest.xml". Celui-ci définit les propriétés principales de l'App : titre, logo, URL relative, mais aussi les permissions qui seront réclamées par celle-ci. Cela permet notamment d'utiliser certains services avancés comme les profils utilisateurs, les fonc-

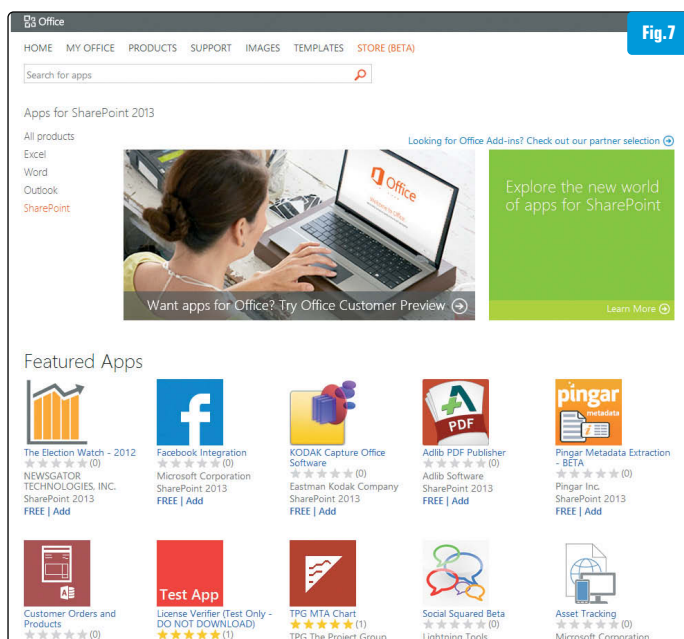


Fig.7

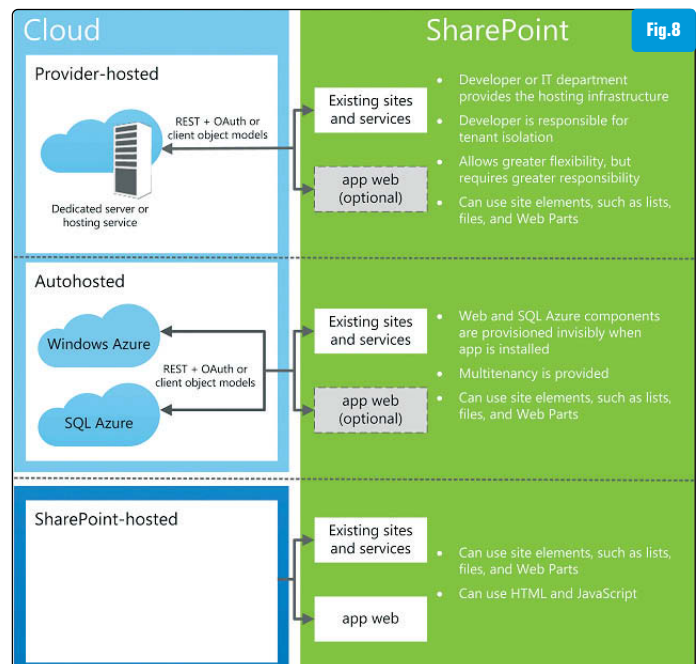


Fig.8

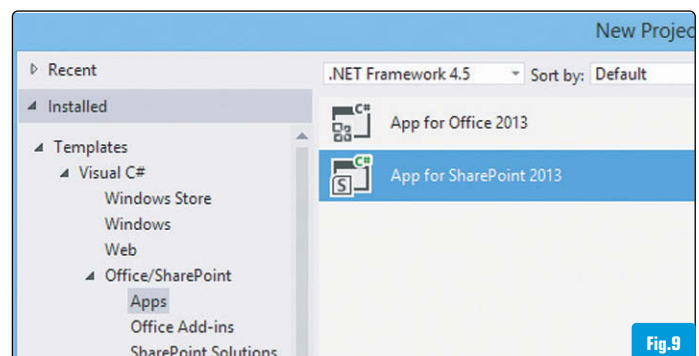


Fig.9

tions sociales ou la recherche avec 4 niveaux d'accès possibles : lecture, écriture, gestion ou contrôle total [Fig.10].

Côté interface, une App sera visible sous forme d'une page à minima, mais aussi de Web Parts "Clientes" (ou "Parts") ou d'éléments dans les menus (« Custom Actions »).

> Développer pour le Cloud et dans le Cloud

Microsoft a poussé encore plus loin le principe du Cloud puisqu'il propose une App pour Office 365 Preview dénommée "Napa Office 365 Developments Tools" qui offre un mini IDE directement accessible depuis le navigateur. Vous pouvez ainsi développer une application pour SharePoint ou Office en ligne, la déployer et la tester depuis une instance Office 365. Cet outil est actuellement disponible à l'essai : il vous suffit de créer un compte développeur sur Office 365 Preview en vous inscrivant depuis l'adresse suivante <http://bit.ly/O365Dev> et en activant l'application Napa.

Vous aurez le choix entre 4 types d'applications : SharePoint, Task Pane pour Office, Excel et Mail [Fig.11].

On bascule ensuite sur l'éditeur qui nous permet d'ajouter des images, modifier le code Javascript et le contenu de la page principale ou des Web Parts. J'ai par exemple modifié ici le code Javascript pour afficher les listes courantes en ayant préalablement modifié ma page "Default.aspx" pour déclarer une balise qui contiendra le rendu final. Vous noterez que jQuery est présent par défaut dans les pages [Fig.12].

J'ai aussi modifié le logo de l'application qui apparaîtra dans le Store et ajouté une image pour la page d'accueil.

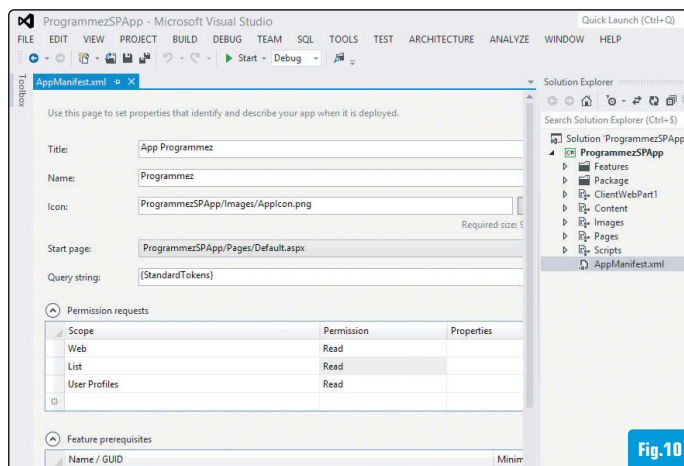


Fig.10

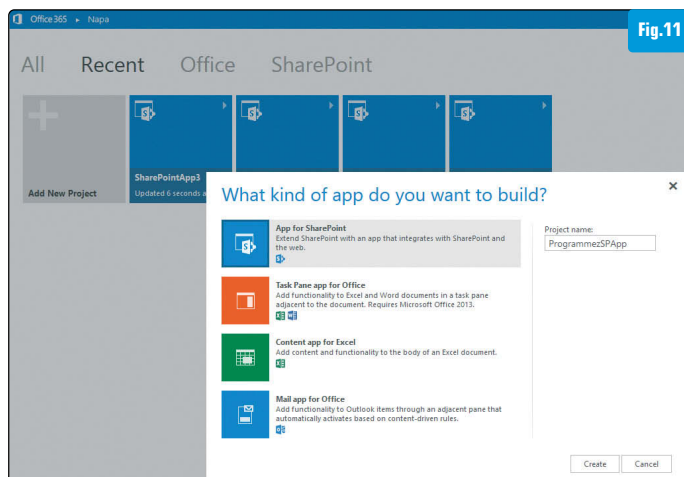


Fig.11

Il suffit ensuite de tester l'application via le "Run Project" situé dans le menu des actions en bas de page [Fig.3].

On obtient alors notre application en tant que lien dans le menu de navigation [Fig.14].

De plus, il est possible d'ouvrir le projet depuis Visual Studio 2012 qui téléchargera tout le projet et vous laissera poursuivre sur votre poste et l'intégrer à votre gestionnaire de code source.

> Conclusion

Nous avons abordé dans cet article les plus gros changements pour les développeurs SharePoint, avec notamment les grandes tendances de fond que sont le Cloud et le magasin d'applications. Malgré tout, les développeurs actuels resteront en terrain connu, Microsoft ayant misé sur la consolidation de l'existant en apportant diverses améliorations, dont un focus tout particulier sur la partie client (modèle objet client, API REST...) afin de permettre une intégration toujours plus aisée avec d'autres environnements. Il ne vous reste plus qu'à essayer par vous-même et peut-être réaliser la future top App pour SharePoint !



Gaëtan Bouveret – Consultant Senior SharePoint MVP SharePoint
Infinite Square / gbouveret@infinitesquare.com
Blog: <http://www.sharepointofview.fr/gat>

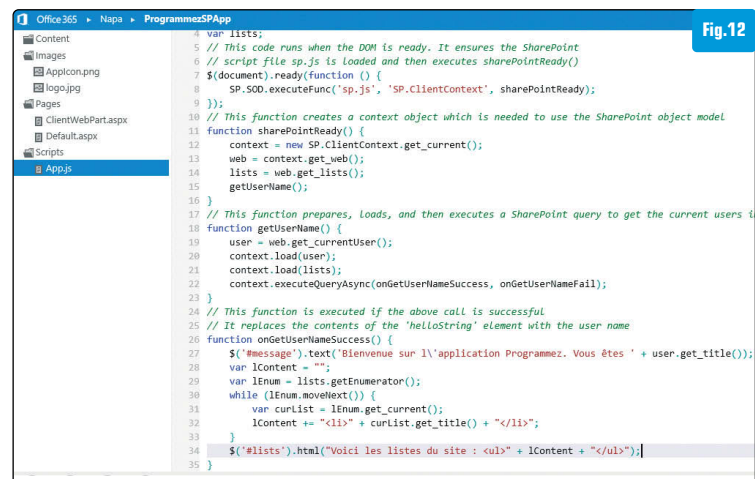


Fig.12

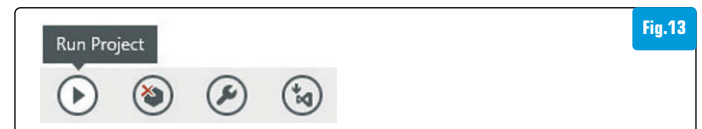


Fig.13

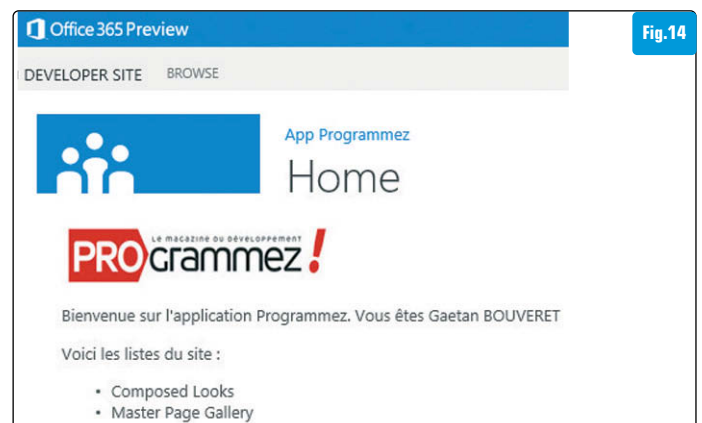


Fig.14

Utilisation de la technologie NFC au sein d'une application Android

Apparue il y a un peu moins d'une dizaine d'années, la technologie NFC commence à connaître un essor certain au sein du monde des smartphones et plus particulièrement de la plateforme Android. Avec des cas d'usage n'ayant de limite que l'imagination des développeurs, il y a fort à parier que cette technologie va se généraliser dans les années à venir et être de plus en plus présente dans notre quotidien. Cet article introductif se propose de montrer comment mettre en œuvre le NFC en lecture et en écriture au sein d'une application Android.

QU'EST-CE QUE LE NFC ?

Le NFC, ou Near Field Communication pour les adeptes de la langue de Shakespeare, est une technologie de communication sans contact qui a le vent en poupe depuis quelque temps maintenant. Dès 2004, de nombreux acteurs industriels du secteur, tels Nokia, Intel ou Visa pour ne citer qu'eux, se sont réunis au sein du NFC Forum afin de spécifier formellement la technologie.

Basé sur la technologie RFID (Radio Frequency Identification), le NFC utilise de fait la bande de fréquence standardisée du RFID à savoir 13.56 MHz (ISO 14443). D'une portée théorique d'un peu plus de 10 cm, le NFC est en pratique utilisé pour des usages à moins de 5 cm avec une vitesse de transfert plutôt faible (moins de 424 kbps). Cette dernière devant être prise en compte lors de l'élaboration de nouveaux use cases NFC.

Concrètement, on distingue 2 types d'appareils NFC : les readers / writers et les Tags. Les smartphones étant les principaux appareils de type readers / writers, la spécification NFC a été surtout pensée pour eux. Les Tags sont quant à eux des éléments passifs composés d'une puce et d'une antenne alimentés par le champ magnétique généré par un reader / writer.

Le NFC Forum définit 4 types de Tags :

- Le Type 1 avec comme principal exemple les cartes Innovision Topaz.
- Le Type 2 tels que les Mifare Ultralight présents partout dans le Monde.
- Le Type 3 tels que les Felica poussés par Sony et présents seulement au Japon.
- Le Type 4 tels que les Mifare Desfire.

La capacité de ces Tags reste restreinte variant de 48 octets à 4 ko. Les exemples les plus simples de ces Tags étant les stickers, les cartes ou les posters avec des usages pouvant aller de la découverte d'URL à l'échange de cartes professionnelles par exemple. Les spécificités fonctionnelles du NFC et un nombre de use cases n'ayant de limite que l'imagination des développeurs font de la technologie le lien parfait entre le monde réel et le monde virtuel, devenant de facto la technologie d'avenir pour le fameux « Internet of Things », dont bon nombre sont promis à un avenir brillant.

FORMAT D'ÉCHANGES

Comme souvent avec des technologies en devenir, les normes de communication ont tendance à se multiplier devant la volonté de

certaines acteurs d'imposer leur propre vision. De fait, il existe déjà 5 normes de communication pour le NFC, soit :

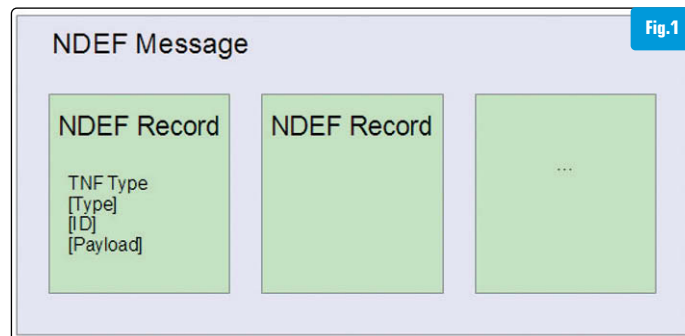
- ISO 14443 - 3A
- ISO 14443 - 3B
- Felica (poussée par Sony, de code JIS 6319-4, supportée par le NFC Forum bien que non standardisée)
- ISO 15693 (non supportée par le NFC Forum)
- ISO-DEP (ISO 14443-4)

Dans un souci de faciliter la communication entre appareils NFC et d'abstraire les clients des spécificités inhérentes à ces différentes normes, le NFC Forum a défini un format d'échange standardisé. Nommé NDEF pour NFC Data Exchange Format, ce format n'est néanmoins pas utilisable avec toutes les normes de communication détaillées précédemment. Ainsi, il s'avère nécessaire dans certains cas de travailler directement sur les données brutes échangées en s'appuyant sur le protocole sous-jacent. Hormis cette problématique, le format NDEF se révèle très pratique pour manipuler simplement les données échangées via NFC.

Le format NDEF [Fig.1] définit un message NDEF correspondant aux données transmises lors d'un échange NFC. Ce message englobe plusieurs enregistrements de type NDEF record. Chacun de ces enregistrements définit les champs suivants :

- Un champ TNF (Type Name Format) indiquant le type de format utilisé.
- Un champ Type permettant d'identifier le type de contenu du Payload.
- Un champ ID permettant éventuellement d'identifier de manière unique un enregistrement.
- Le Payload qui est le contenu à proprement parler du record.

On constate sur la [Fig.1] qu'un message NDEF peut potentiellement contenir plusieurs enregistrements. Ainsi, il se peut que le contenu



Message au format NDEF

entier du message ne soit pas présent dans l'unique Payload du premier enregistrement ou bien que plusieurs informations soient contenues au sein d'un même message.

AU SEIN D'ANDROID

La plateforme Android fournit un support NFC digne de ce nom depuis Gingerbread en version 2.3.3. Ce support propose une API relativement complexe mais puissante, donnant accès aux développeurs à tout ce qu'il faut pour la lecture / écriture sur des Tags, ainsi que pour le formatage des données NDEF. De fait, l'utilisation du NFC au sein d'une application nécessite de déclarer un support minimum du SDK en version 10 au sein du manifest comme suit :

```
<uses-sdk android:minSdkVersion="10" />
```

En outre, l'utilisation du NFC implique la déclaration de la permission dédiée android.permission.NFC au sein du manifest :

```
<uses-permission android:name="android.permission.NFC" />
```

Le filtrage au niveau du Google Play Store, pour rendre une application uniquement visible aux périphériques supportant le NFC, se faisant via l'ajout suivant au manifest :

```
<uses-features android:name="android.hardware.nfc" android:required="true" />
```

Comme expliqué précédemment, tous les Tags et appareils NFC ne supportent pas le format NDEF. En effet, s'il existe déjà plus de 5 milliards de Tags NFC déployés, la plupart d'entre eux ne contiennent pas de messages NDEF. Pour ces Tags spécifiques, Android propose le package android.nfc.tech exposant des classes qui offrent des fonctionnalités spécifiques aux diverses technologies NFC. Le mapping existant entre ces dernières et les classes du package susmentionné est présenté au sein du tableau de la [Fig.2].

Toutes ces classes représentant des implémentations spécifiques de l'interface TagTechnology, l'API peut être étendue en implémentant cette dernière pour fournir le support d'une nouvelle technologie de Tag. Il est bon également de noter que les périphériques Android dotés d'un capteur NFC se doivent de fournir les implémentations de TagTechnology pour les classes suivantes : NfcA, NfcB, NfcF, NfcV, IsoDep, Ndef et NdefFormattable. Les implémentations pouvant exister pour les autres technologies ne sont qu'optionnelles et peuvent varier d'un périphérique à l'autre.

Enfin, Ice Cream Sandwich (version 4.0) aura également apporté une nouveauté majeure aux fonctionnalités NFC de la plateforme avec Android Beam. Cette dernière API permet le push de messages NDEF, ce qui autorise la simulation du mode passif du NFC pour un appareil Android en le transformant de facto en Tag, à la restriction près qu'il reste alimenté par sa batterie et non par le champ magnétique d'un autre appareil. Ainsi, Beam vient combler l'absence du mode émulation du capteur NFC au sein de l'API Android.

Fig.2					
Tag Technologie	ISO 14443 - 3A	ISO 14443 - 3B	Felica	ISO 15693	ISO 14443 - 4
Implémentation de TagTechnology	NfcA	NfcB	NfcF	NfcV	IsoDep

Mapping entre les technologies de Tags et leurs classes Android équivalentes

DÉTECTION D'UN TAG

Avant de pouvoir interagir avec un Tag NFC, il est nécessaire que le système Android soit informé que notre application permet de répondre à la détection générale d'un tel Tag. En fonction du type de message contenu au sein du Tag, l'action de l'Intent envoyée par l'OS sera différente. Dans l'ordre des priorités, le mécanisme de dispatch des Tags [Fig.3] teste en premier si le Tag contient un message NDEF correctement formaté auquel cas l'action NDEF_DISCOVERED est envoyée. Dans le cas contraire, si le Tag ne contient pas de message NDEF correct ou qu'il n'est pas possible d'identifier un type MIME ou une URI à partir du message, ou si aucune application ne possède un filtre pour l'action NDEF_DISCOVERED, l'action de l'Intent TECH_DISCOVERED est envoyée. Enfin, au cas où aucune activité ne possède le filtre idoine, un Intent est envoyé avec l'action TAG_DISCOVERED.

Ainsi, le choix du filtre d'Intent associé à une activité se révèle essentiel. En effet, une application ne doit pas être proposée à l'utilisateur si elle ne peut gérer le Tag qui vient d'être détecté. A contrario, en filtrant uniquement sur l'action TAG_DISCOVERED, une application ne sera appelée à la détection d'un Tag que si aucune autre n'est en mesure de répondre aux 2 premières conditions exprimées dans le diagramme de la [Fig.3].

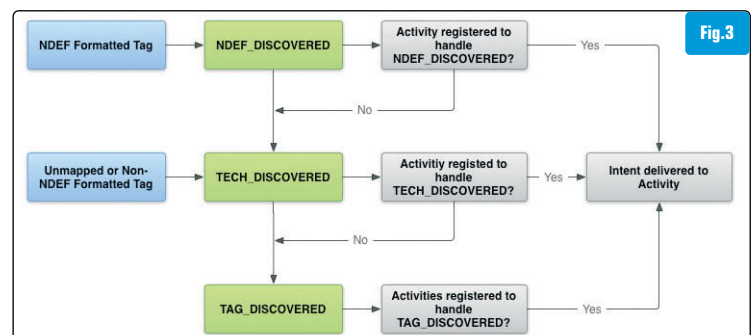
Pour notre application, nous définissons ainsi au sein du manifest une activité NFCTagViewer qui répondra à l'Intent d'action NDEF_DISCOVERED :

```
<activity android:name="NFCTagViewer">
  <intent-filter>
    <action android:name="android.nfc.action.NDEF_DISCOVERED"/>
    <category android:name="android.intent.category.DEFAULT"/>
  </intent-filter>
</activity>
```

LECTURE D'UN TAG

A la détection d'un Tag contenant un message NDEF correctement formé, notre application sera proposée par le système pour le gérer puisque son activité NFCTagViewer le permet. Pour que notre application puisse gérer un Tag lorsqu'elle est au premier plan, il est nécessaire de modifier légèrement la méthode onResume de notre activité NFCTagViewer en activant le dispatch au premier plan. Notons qu'il ne faut pas oublier dans ce cas-là de désactiver ce dispatch dans la méthode onPause de l'activité.

Le contenu de notre activité permettant de détecter et de lire le contenu de Tags contenant des messages NDEF bien formés est le suivant :



Mécanisme de dispatch des Tags

```
public class NFCTagViewer extends Activity {
    private static final String TAG = "NFCTagViewer";
    private NfcAdapter nfcAdapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.tag_viewer);
        nfcAdapter = NfcAdapter.getDefaultAdapter(getApplication()
Context());

        if (nfcAdapter == null || !nfcAdapter.isEnabled()) {
            finish();
            return;
        }
    }

    @Override
    protected void onResume() {
        super.onResume();
        PendingIntent pendingIntent = PendingIntent.getActivity(this, 0,
            new Intent(this, this.getClass()).addFlags(Intent.FLAG_
ACTIVITY_SINGLE_TOP), 0);
        IntentFilter[] filters = null; // filtres de Tags
        String[][] techListArray = null; // filtres de Technologies de Tags
        nfcAdapter.enableForegroundDispatch(this, pendingIntent, filters,
techListArray);
    }

    @Override
    protected void onPause() {
        nfcAdapter.disableForegroundDispatch(this);
        super.onPause();
    }

    @Override
    public void onNewIntent(Intent intent) {
        setIntent(intent);
        resolveIntent(intent);
    }

    private void resolveIntent(Intent intent) {
        String action = intent.getAction();

        if (NfcAdapter.ACTION_NDEF_DISCOVERED.equals(action)) {
            Parcelable[] rawMsgs = intent.getParcelableArrayExtra(Nfc
Adapter.EXTRA_NDEF_MESSAGES);
            if (rawMsgs != null) {
                NdefMessage[] msgs = new NdefMessage[rawMsgs.length];
                for (int i = 0; i < rawMsgs.length; i++) {
                    msgs[i] = (NdefMessage) rawMsgs[i];
                }

                parseMsgs(msgs);
            } else {
                Log.e(TAG, "Unknown tag type");
            }
        } else {
```

```
            Log.e(TAG, "Unknown intent " + intent);
            finish();
            return;
        }
    }

    private void parseMsgs(NdefMessage[] msgs) {
        for (NdefMessage msg : msgs) {
            for (NdefRecord record : msg.getRecords()) {
                // parser de txts
                TextRecord textRecord = TextRecord.parse(record);

                if (textRecord != null) {
                    String content = textRecord.getContent();
                } else {
                    // support d'autres types à implémenter
                }
            }
        }
    }
}
```

Le Tag découvert contient des messages NDEF correctement formés. Comme expliqué précédemment, un message NDEF, modélisé par la classe `NdefMessage` au sein de la plateforme, peut contenir plusieurs enregistrements de type `NdefRecord`. Notre activité se charge donc de récupérer les données provenant du Tag passées par l'OS à travers l'Intent. On itère ensuite sur les messages NDEF et les enregistrements qu'ils contiennent. Pour traiter les données textuelles des enregistrements, on définit un objet `TextRecord` possédant la méthode `parse` chargée de parser le contenu de l'enregistrement et de créer l'objet `TextRecord` correspondant comme suit :

```
public static TextRecord parse(NdefRecord record) {
    TextRecord txtRecord = null;

    try {
        byte[] payload = record.getPayload();
        // décodage d'un payload de type texte
        String textEncoding = ((payload[0] & 0200) == 0) ? "UTF-8" : "UTF-16";
        int languageCodeLength = payload[0] & 0077;
        String languageCode = new String(payload, 1, languageCode
Length, "US-ASCII");
        String text = new String(payload, languageCodeLength + 1, pay
load.length - languageCodeLength - 1, textEncoding);
        txtRecord = new TextRecord(languageCode, text);
    } catch (UnsupportedEncodingException e) {
        txtRecord = null;
    }

    return txtRecord;
}
```

Pour des Tags de type URI ou SMART_POSTER, la démarche resterait globalement identique avec la création d'objets spécifiques de type `UriRecord` et `SmartPosterRecord` au sein desquels la méthode `parse` implémenterait le traitement particulier chargé de décoder les

payloads spécifiques à ces formats. Des exemples de décodage de ces formats de données peuvent être trouvés au sein du SDK Android.

ECRITURE SUR UN TAG

Maintenant que nous avons détaillé la réception et la lecture d'un Tag NFC, nous allons utiliser l'API Android pour écrire et envoyer des données vers un Tag. La partie de détection d'un Tag reste la même que pour la partie précédente. La différence majeure se situe dans la méthode `onNewIntent` au sein de laquelle nous allons écrire un message NDEF sur le Tag détecté :

```
@Override
public void onNewIntent(Intent intent) {
    setIntent(intent);
    NdefMessage msg = creerMessage("Hello Programmez !", true);
    writeMessage(intent, msg);
}

private void writeMessage(Intent intent, NdefMessage msg) {
    Tag tag = intent.getParcelableExtra(NfcAdapter.EXTRA_TAG);
    Ndef ndef = Ndef.get(tag);
    try {
        ndef.connect();
        ndef.writeNdefMessage(msg);
        ndef.close();
    } catch (Exception e) {
        Log.e(TAG, "Erreur à l'écriture");
    }
}

private NdefMessage creerMessage(String text, boolean encodeInUtf8) {
    NdefRecord[] records = new NdefRecord[1];
    records[0] = TextRecord.createTextRecord(text, Locale.FRENCH,
    encodeInUtf8);
    return new NdefMessage(records);
}
```

Le message NDEF envoyé contient simplement des données textuelles et est créé au sein de la méthode `creerMessage`. L'objet `NdefMessage` créé contient donc un seul record créé via la méthode utilitaire `createTextRecord` de notre classe `TextRecord` et dont le contenu est le suivant :

```
public static NdefRecord createTextRecord(String text, Locale
locale, boolean encodeInUtf8) {
    byte[] langBytes = locale.getLanguage().getBytes(Charset.for
Name("US-ASCII"));
    Charset utfEncoding = encodeInUtf8 ? Charset.forName("UTF-8")
: Charset.forName("UTF-16");
    byte[] textBytes = text.getBytes(utfEncoding);
    int utfBit = encodeInUtf8 ? 0 : (1 << 7);
    char status = (char) (utfBit + langBytes.length);
    byte[] data = Bytes.concat(new byte[] {(byte) status}, lang
Bytes, textBytes);
    return new NdefRecord(NdefRecord.TNF_WELL_KNOWN, NdefRecord.RTD
_TEXT, new byte[0], data);
}
```

L'écriture du message sur le Tag détecté est réalisée au sein de la méthode `writeMessage`. L'intent passé en paramètre permet de récupérer un objet de type `Tag` à partir duquel nous allons pouvoir effectuer des opérations d'écriture ou de lecture. Dans notre cas, il s'agit d'écriture et puisque l'on souhaite écrire des données NDEF on utilise la méthode statique `get` de la classe `Ndef` qui renvoie un objet `Ndef` si et seulement si la technologie est supportée par le Tag. La suite consiste à se connecter, à écrire le message précédemment généré avant finalement de fermer le flux d'écriture du Tag.

SIMULER LA RÉCEPTION D'UN TAG

Lors du développement de fonctionnalités utilisant le NFC, il peut s'avérer intéressant de simuler la réception d'un Tag à des fins de débogage notamment. Pour cela, nous créons un Intent avec une action `ACTION_TAG_DISCOVERED` au sein duquel nous plaçons un message NDEF :

```
Intent intent = new Intent(NfcAdapter.ACTION_TAG_DISCOVERED);
NdefMessage[] messages = new NdefMessage[1];
messages[0] = creerMessage("Simulation message NDEF", true);
intent.putExtra(NfcAdapter.EXTRA_NDEF_MESSAGES, messages);
startActivity(intent);
```

Il reste ensuite à définir une activité chargée de simuler la réception des données de ce Tag comme détaillé précédemment.

ANDROID BEAM

Grande nouveauté d'Ice Cream Sandwich pour le NFC, Android Beam autorise les échanges P2P simples entre appareils via son mécanisme de push de messages NDEF. Au sein de la plateforme, Beam est mis en œuvre pour des échanges médias sans clic. Ainsi, il suffit d'approcher 2 appareils possédant un capteur NFC pour échanger un contact, une page web, une vidéo Youtube ou bien encore une application. La mise en œuvre de Beam se révèle simple et se fait via 2 méthodes proposées par le `NfcAdapter` : `setNdefPushMessage` ou bien `setNdefPushMessageCallback` qui permettent l'envoi de messages NDEF au sein d'une activité devant être au premier plan.

CONCLUSION

Supporté depuis bientôt 2 ans par la plateforme Android, le NFC est une technologie en plein essor promise à un bel avenir. Avec un nombre croissant de périphériques la supportant et des expérimentations sans contact toujours plus nombreuses au sein de villes, il sera possible dans un avenir proche en France d'utiliser son périphérique Android pour effectuer des paiements, comme ticket électronique ou bien pour prendre les transports en commun comme cela peut se faire dans certaines villes américaines.

Outre ces opportunités futures, le développeur peut dès à présent tirer profit du sans contact au sein d'une application pour effectuer des échanges P2P, initier des connexions Bluetooth pour des jeux à plusieurs ou bien même mapper un objet réel sur une URL. Avec des usages quasiment illimités, les développeurs se doivent donc dès maintenant de penser à demain et de booster leurs applications en y intégrant des fonctionnalités NFC.

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

Démystifier le développement des plug-ins d'Eclipse

2^e partie

Dans l'article précédent, nous avons créé un projet plug-in d'Eclipse, ajouté des modèles (templates), pour faciliter le traitement des fichiers XML en Java, et testé le plug-in dans une nouvelle instance d'Eclipse. Dans cette suite, nous allons continuer le plug-in en ajoutant d'autres fonctionnalités.

AJOUTEZ DES CHEAT SHEETS (FEUILLES DE TRICHE) AU PROJET

Une cheat sheet ou feuille de triche est un mini tutoriel interactif, qui s'affiche dans Eclipse, servant comme aide-mémoire, ou comme une aide qui accompagne un plug-in. Pour afficher les cheat sheets disponibles le menu **Help > Cheat Sheets**.

Note : Si pas d'entrée Cheat Sheets sous Help :

Window > Customize Perspective > Commands > cochez la case **Cheat Sheets**.

Dans Eclipse Juno au lieu de Commands, sélectionnez Command Groups Availability.

Il existe dans Eclipse, deux catégories de cheat sheets :

- Une cheat sheet simple qui décrit une tâche «simple», composée normalement de plusieurs étapes, comme par exemple «créer un projet plug-in»
- une cheat sheet composite qui contient plusieurs tâches, divisées éventuellement en plusieurs groupes. Par exemple, la création d'un plug-in Eclipse, qu'on peut décrire en plusieurs tâches simples ; la création d'un projet plug-in, ajouter des extensions, tester le plug-in, l'exporter sous forme d'un fichier .jar, etc.

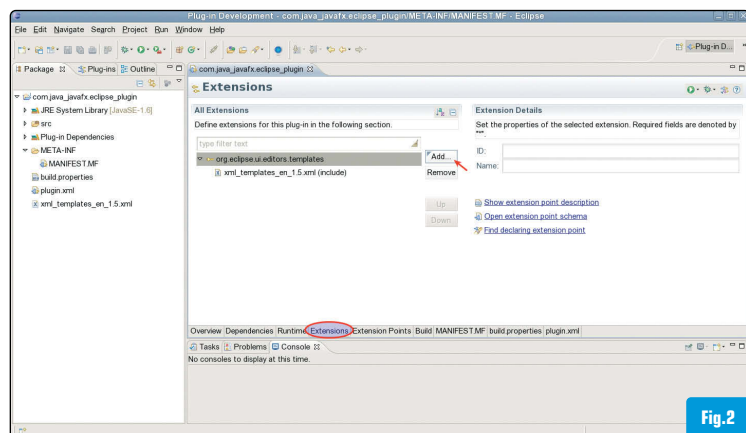
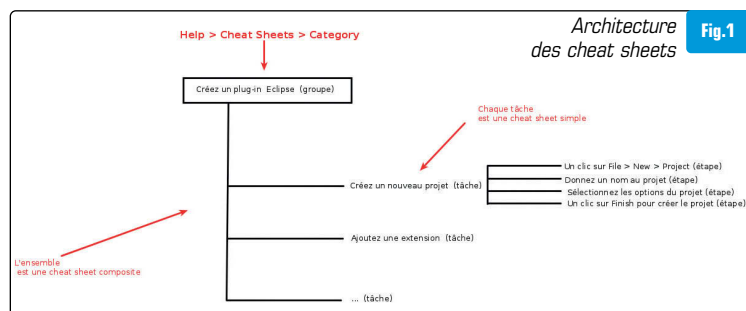


Fig.2

Ajouter une extension

> Présentation des cheat sheets dans Eclipse [Fig.1]

> Comment créer des cheat sheets (simples ou composites)

- Création d'un projet plug-in (déjà fait)
- L'ajout d'une extension pour les cheat sheets
- L'édition des cheat sheets
- Le test des cheat sheets
- L'exportation des cheat sheet (dans le plug-in fini).

> Ajoutez une extension au projet

Dans l'onglet **Extensions** > un clic sur le bouton **Add**

Note : Si l'éditeur spécial de plug-ins n'est pas ouvert > double cliquez sur **MANIFEST.MF** à gauche dans **Package Explorer**.

[Fig.2]

Décochez **Show only extension points from the required plug-ins**

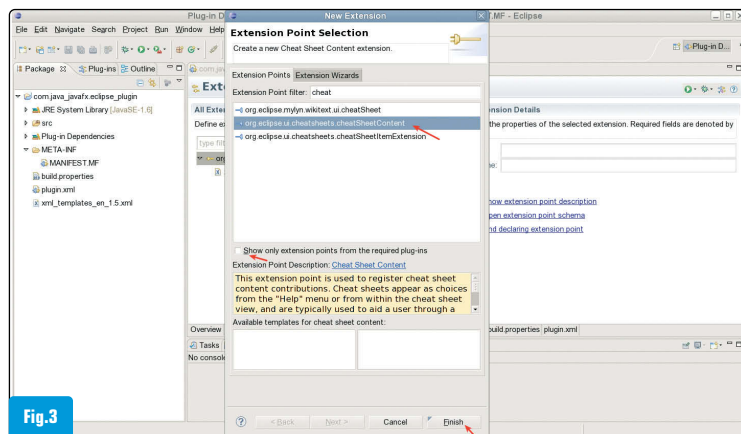


Fig.3

Sélectionner un point d'extension

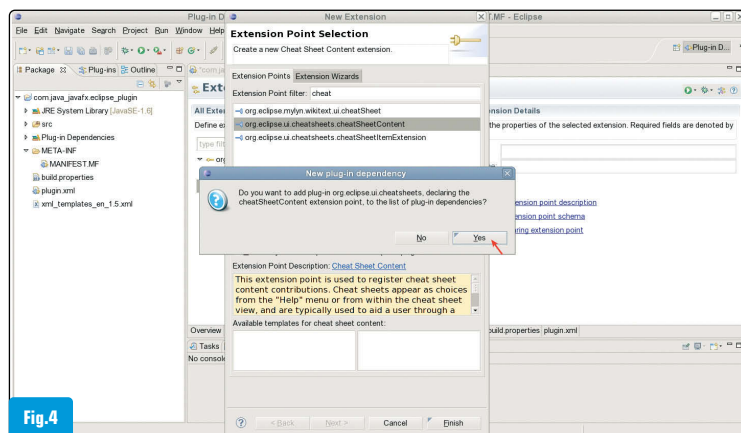


Fig.4

Confirmer les dépendances

Entrez le mot cheat dans le champ **Extension Point filter**
Sélectionnez dans la liste **org.eclipse.ui.cheatsheets.cheatSheetContent**. **Finish** [Fig.3]

Confirmez l'ajout de dépendances (Confirm adding plug-in dependency). Un clic sur **Yes** [Fig.4]

> Ajoutez une catégorie de cheat sheet

Note : Cette catégorie s'affichera sous **Help > Cheat Sheets** [Fig.5]

Pour ajouter une catégorie > Un clic droit sur l'extension, que nous venons de créer

org.eclipse.ui.cheatsheets.cheatSheetContent > New > category [Fig.6]. Donnez un nom de votre choix à cette catégorie. Il est aussi possible de personnaliser l'identificateur **id** [Fig.7]

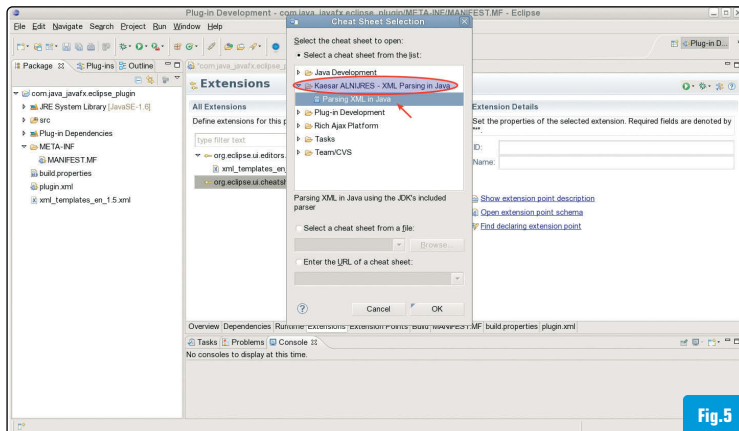


Fig.5

L'affichage d'une catégorie

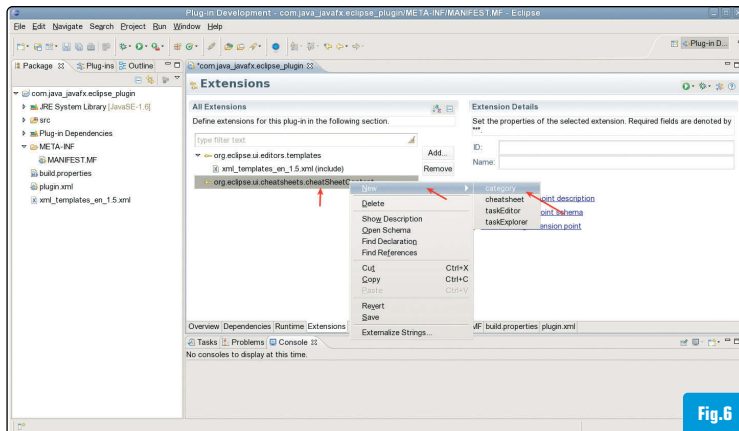


Fig.6

Ajouter une catégorie

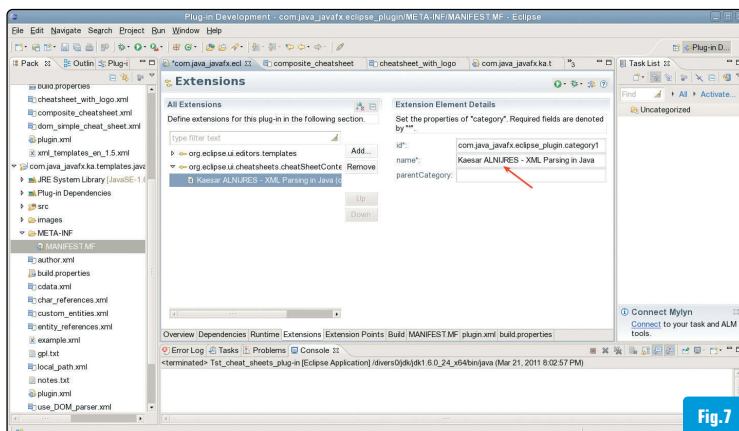


Fig.7

Personnaliser la catégorie

> Ajoutez une cheat sheet composite

Sélectionnez le projet dans **Package Explorer**

Dans le menu **File > New > Other**

Dans la boîte de dialogue **New**

Sous la catégorie **User Assistance >** sélectionnez **Cheat Sheet > Next** [Fig.8]

Dans la boîte de dialogue **New Cheat Sheet**

Vérifiez que le projet est sélectionné

Cochez **Composite Cheat Sheet**

Entrez un nom pour le fichier de la cheat sheet dans **File name** avec l'extension **.xml**. **Finish** [Fig.9]

L'éditeur spécial cheat sheet est ouvert

Sélectionnez **Title** (à gauche), entrez le titre principal de la cheat sheet dans le champ **Name**

L'image présente aussi l'affichage du titre (**Help > Cheat Sheets**). Le nom de la cheat sheet composite est affiché sous la catégorie entrée précédemment [Fig.10].

> Ajoutez un groupe

Group(1) est le nom d'un objectif global décrit dans la cheat sheet composite. Entrez une **introduction(2)** et une **conclusion(3)**.

L'image présente ces composants, affichés dans une cheat sheet d'Eclipse et leurs emplacements dans l'éditeur [Fig.11].

> Ajoutez des tâches

Quand on ajoute un groupe, une tâche est ajoutée automatiquement dans la cheat sheet. Il faut la personnaliser. Un clic sur **task**

Entrez son nom **Name (1)**, ajoutez une **introduction(2)** et une **conclusion(3)**. Le plus important ici, est d'associer une cheat sheet

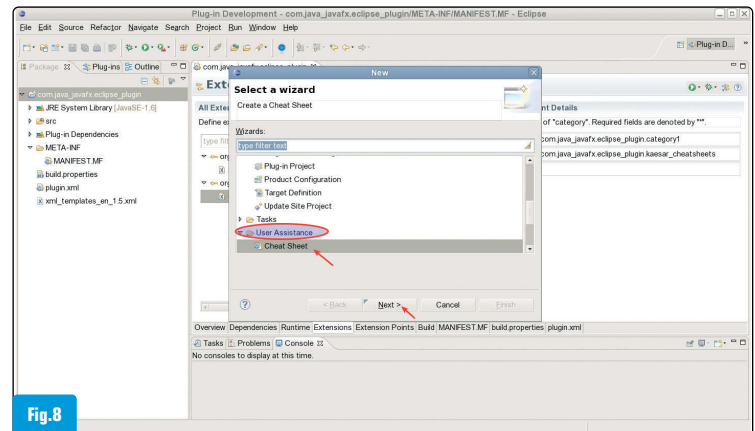


Fig.8

Ajouter une Cheat Sheet composite

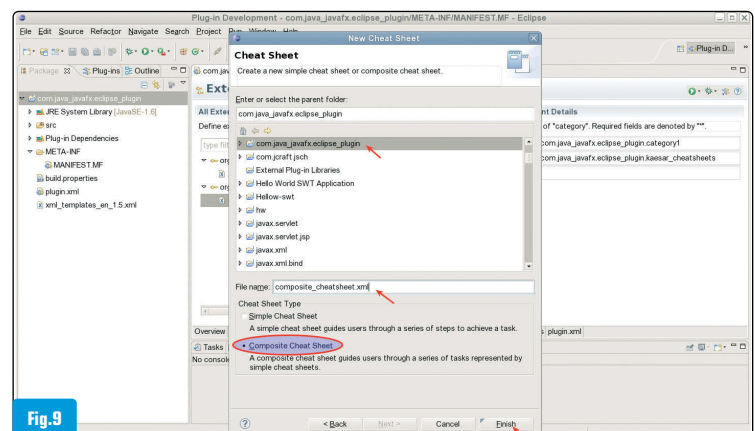


Fig.9

Nommer le fichier composite

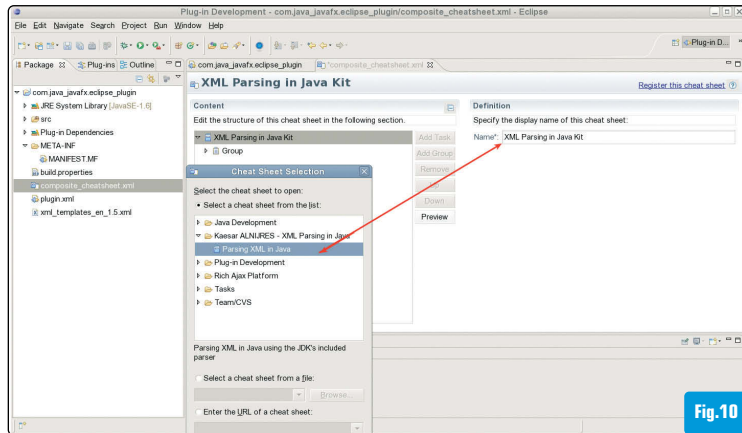


Fig.10

Nom de la cheat sheet

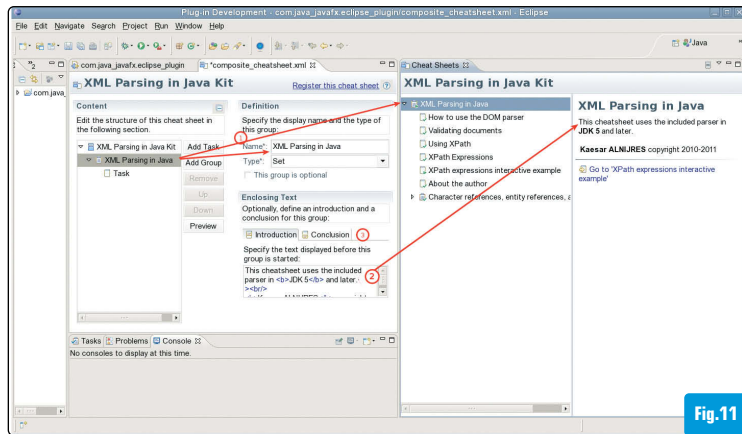


Fig.11

Editer le groupe

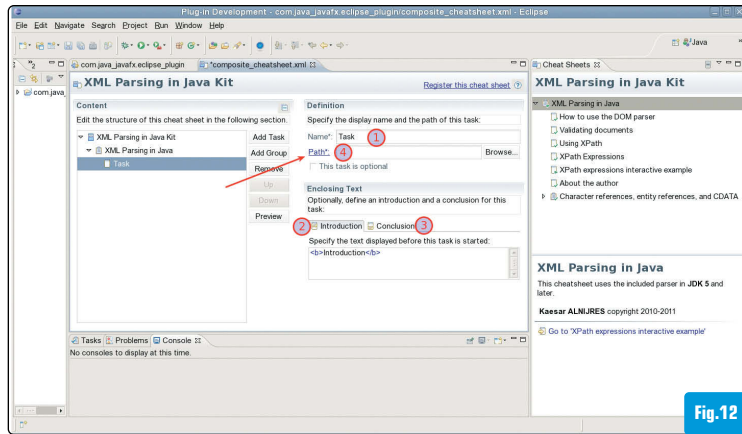


Fig.12

Editer une tâche

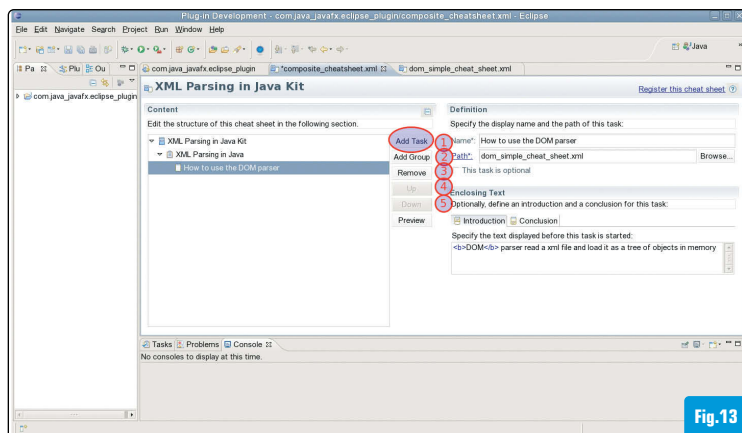


Fig.13

Editeur de cheat sheets composites

simple à cette tâche. Pour attacher une simple cheat sheet, on utilise le lien **Path(4)**. Ce lien ouvre l'éditeur de simple cheat sheets et permet d'associer la tâche en cours à un fichier (cheat sheet simple) [Fig.12]

> Comment ajouter des tâches

Dans l'éditeur cheat sheet composite un clic sur le bouton **Add Task(1)**

Autres fonctionnalités de l'éditeur :

Ajouter des groupes, en cliquant sur **Add Group(2)**

Supprimer une tâche ou un groupe avec **Remove(3)**

Monter une tâche avec **Up(4)** ou la descendre avec **Down(5)** [Fig.13]

Note : Pour avoir un aperçu d'une cheat sheet (simple ou composite), un clic sur le bouton **Preview**.

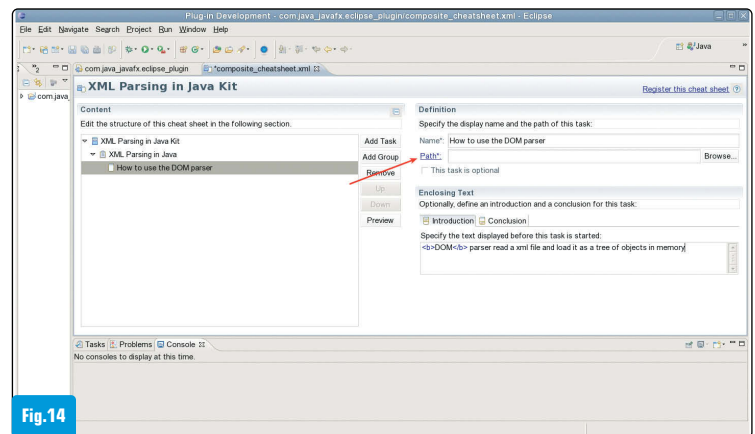


Fig.14

Associer un fichier à une cheat sheet simple

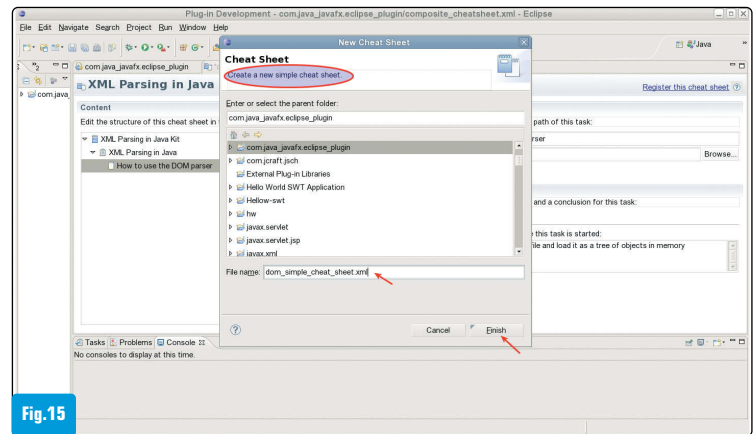


Fig.15

Nommer le fichier

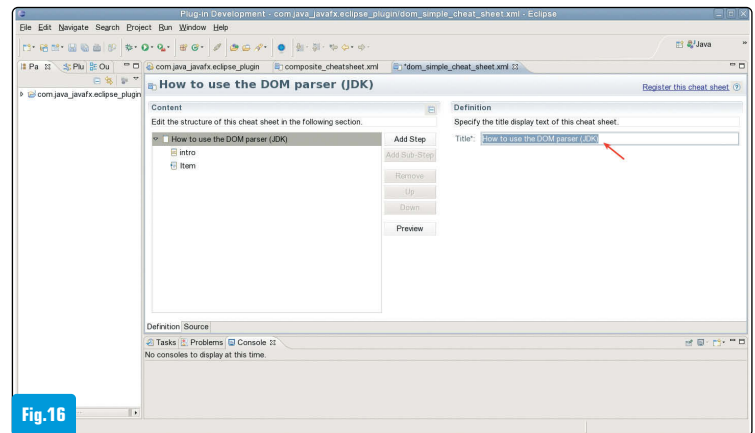


Fig.16

Editeur simple cheat sheet

> Associez un fichier à une tâche

Comme mentionné précédemment, un clic sur le lien **Path**, ouvre l'assistant de simple cheat sheets [Fig.14]

Dans le dialogue **New Cheat Sheet**

Donnez un nom au fichier dans **File name** avec l'extension **.xml**

Quand on clique sur **Finish** l'éditeur de simple cheats s'ouvre [Fig.15]

Dans l'éditeur simple cheat sheets

Entrez un titre à cette tâche dans **Title** et une petite description (intro) [Fig.16]

> Ajoutez une étape à la tâche

Une tâche **Task** est composée de plusieurs étapes, chacune peut avoir des sous-étapes **Sub-Step**.

La première étape est ajoutée automatiquement, lors de la création de la simple cheat sheet

Commencez par donner un titre à l'étape **Title(1)**. Donnez une petite description dans **Body(2)**.

Note :

Il est possible d'ajouter une commande (**Command**) à l'étape, permettant de faire exécuter par Eclipse

Une commande, comme par exemple lancer l'assistant de nouveaux projets. Cette commande est exécutée en cliquant sur un lien à l'intérieur d'une cheat sheet. Pour ajouter une commande, il faut la sélectionner parmi les commandes prédéfinies en cliquant sur le bouton **Browse(3)**

Attention :

Une étape ou Step est appelée Item (balise ou tag <item>) dans le fichier .xml d'une simple cheat sheet [Fig.17 et 18]

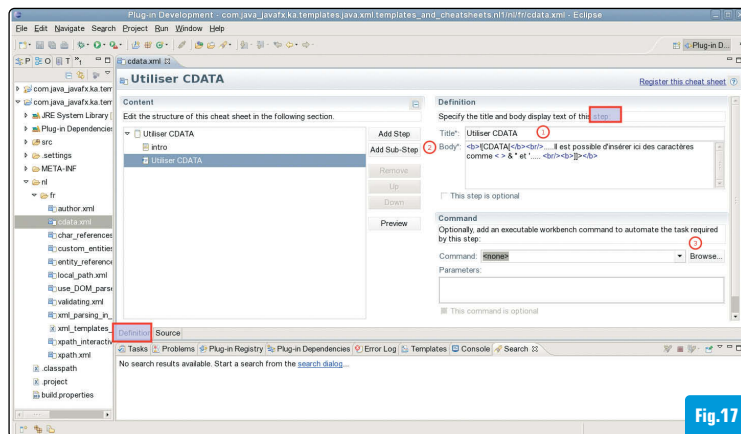


Fig.17

Step dans l'éditeur

> Ajoutez d'autres étapes (Step ou Item) à la tâche courante

Dans l'éditeur (simple cheat sheets) un clic sur **Add Step** [Fig.19]

> Autres fonctions de l'éditeur

Dans l'éditeur de simple cheat sheets, il est possible de :

Ajouter des sous-étapes **Add Sub Step(1)**

Supprimer une étape ou une sous-étape **Remove(2)**

Monter une étape ou une sous-étape **Up(3)** ou les descendre **Down(4)**

Avoir un aperçu **Preview(5)** [Fig.20]

> L'aperçu d'une tâche (simple cheat sheet)

Un clic sur **Preview** ouvre la cheat sheet courante [Fig.21]

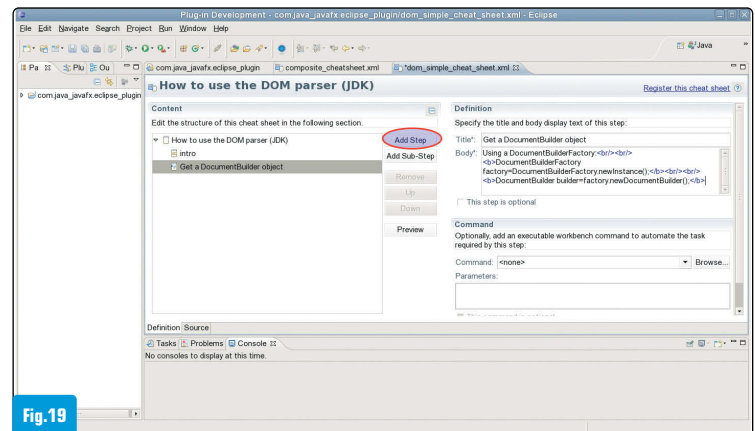


Fig.19

Ajouter une étape

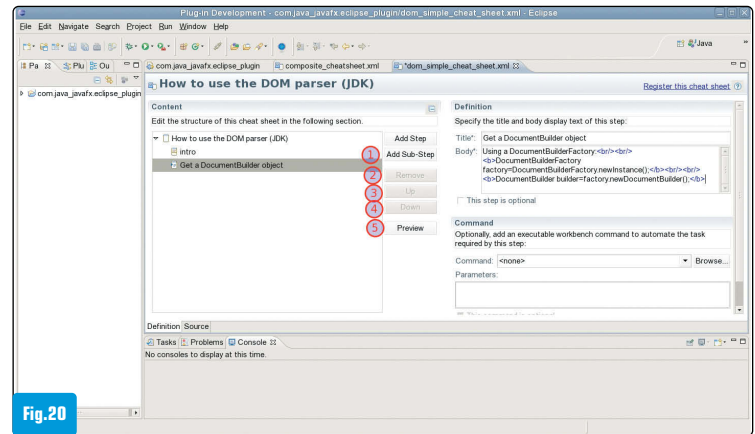


Fig.20

Fonctions de l'éditeur simple cheat sheet

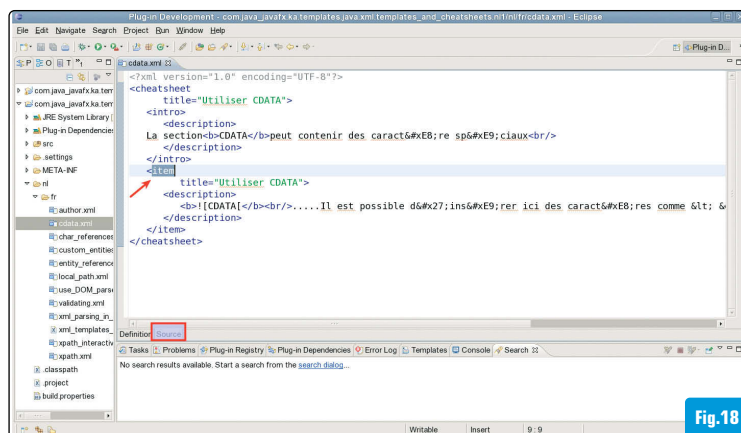


Fig.18

Step est appelée «item» dans le fichier .xml

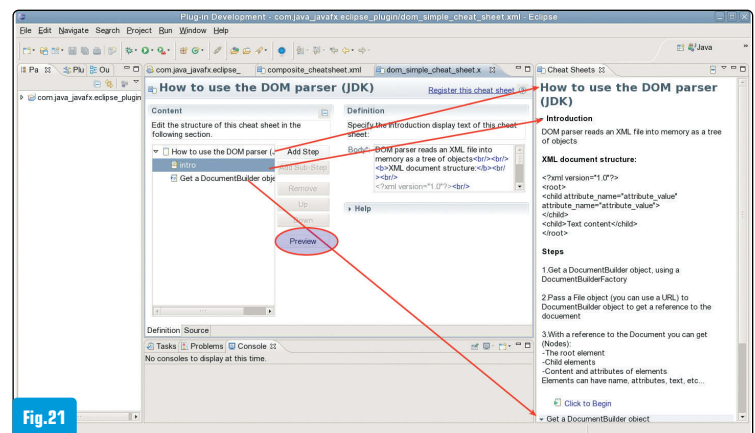


Fig.21

Aperçu d'une tâche

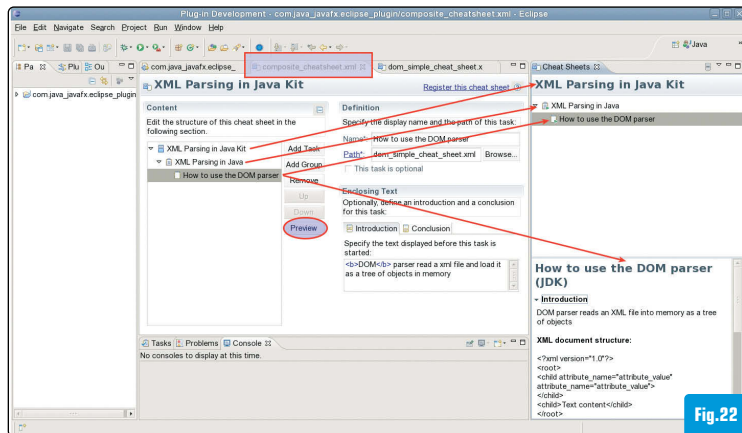


Fig.22

Aperçu composite

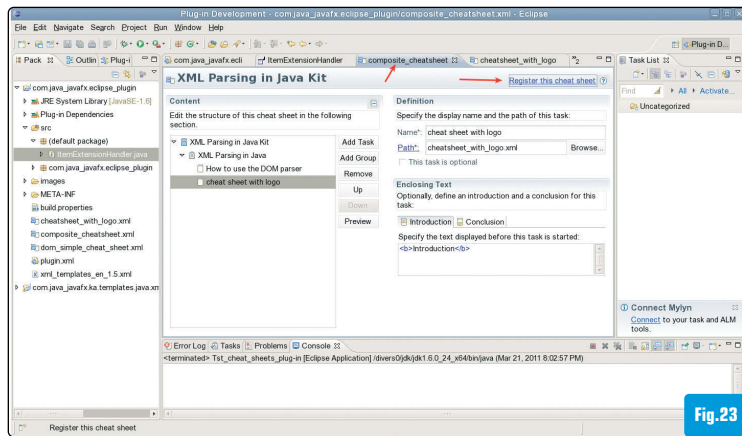


Fig.23

Enregistrer la cheat sheet 1

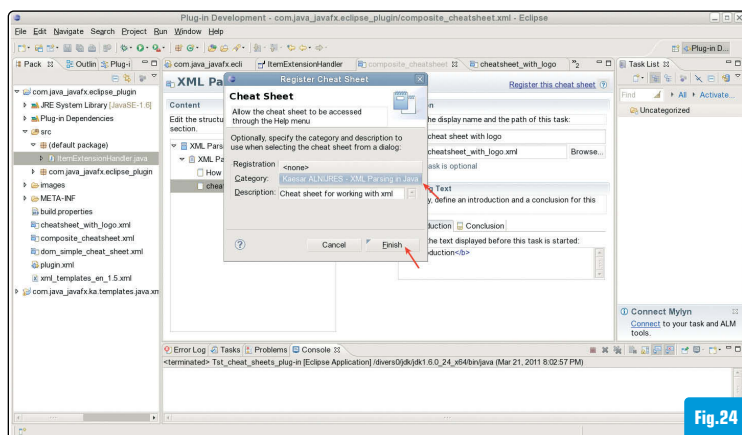


Fig.24

Enregistrer la cheat sheet 2

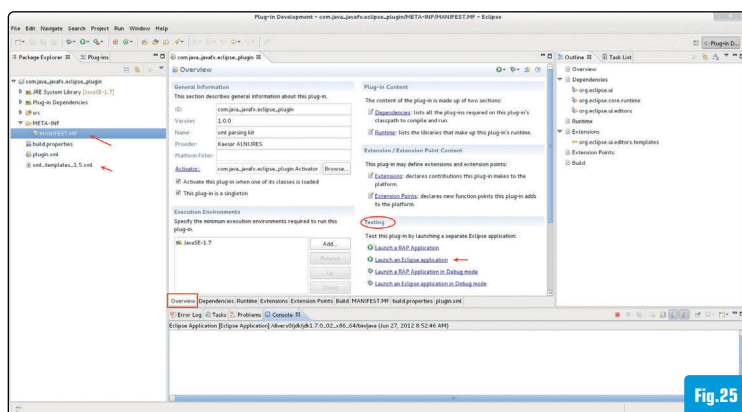


Fig.25

Tester le plug-in

> Aperçu d'une cheat sheet composite

Sélectionnez la cheat sheet composite dans l'éditeur actif (un clic sur sa barre de titre) > un clic sur **Preview** [Fig.22]

> Enregistrez la cheat sheet composite dans le plug-in

Rendez la cheat sheet active, en cliquant sur sa barre de titre. Un clic sur le lien **Register this cheat sheet** [Fig.23].

Dans la boîte de dialogue **Register Cheat Sheet**

Sélectionnez la catégorie créée là-haut, en utilisant la flèche de **Category**. Entrez une petite **Description**. Finish [Fig.24]

> Testez le plug-in

Dans l'onglet **Overview** de l'éditeur de **MANIFEST.MF** > Cliquez sur **Launch Eclipse application**. Une nouvelle instance d'Eclipse est lancée. Le menu **Help > Cheat Sheets** > Sélectionnez votre cheat sheet (le nom donné à la catégorie) [Fig.25 et 26]. La cheat sheet est affichée dans la fenêtre ou View de **Cheat Sheets** [Fig.27].

Note : Insérer une image dans une cheat sheet nécessite une technique spéciale que j'aborderai dans la prochaine partie.

Kaeser Alnires

Développeur Java - Leader jug-cergy - <http://www.java-javafx.com>

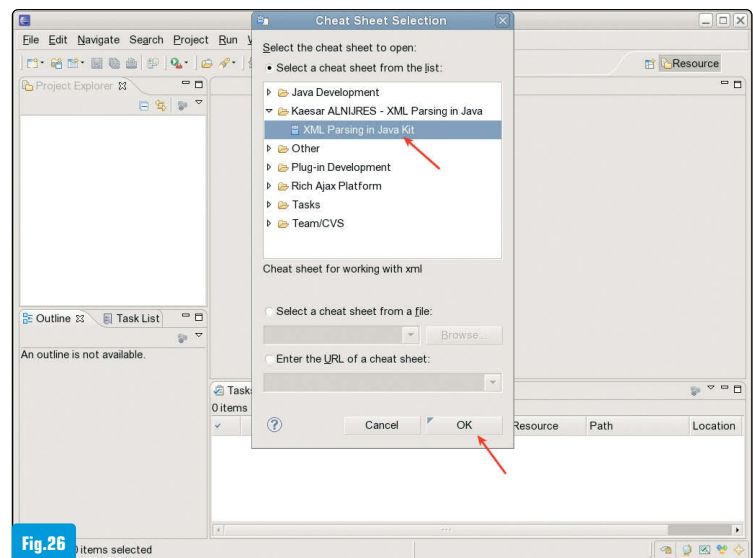


Fig.26

Items selected

Affichage final 1

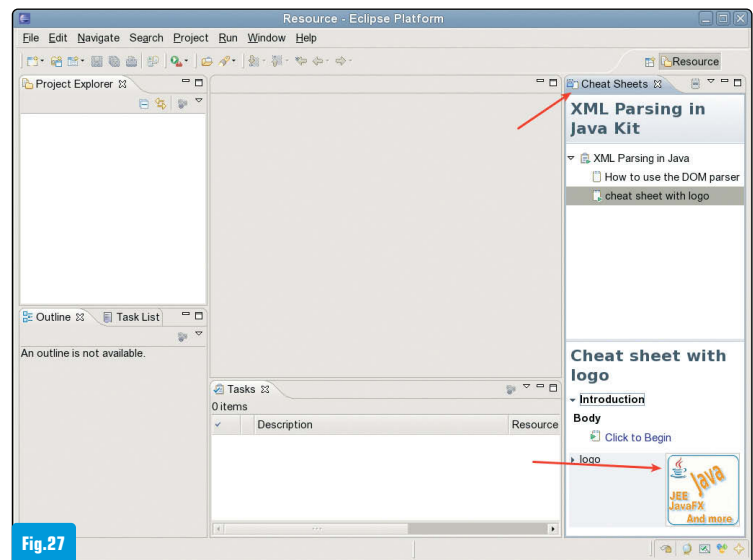


Fig.27

Affichage final 2

Optimisation des performances sur Windows Phone

2^e partie

Les performances jouent un rôle déterminant lors de la première exécution d'une application par un nouvel utilisateur. Un lancement rapide et une exécution fluide de votre application vous garantiront une bonne première impression, et vous aideront à décrocher les 5 étoiles sur le marketplace !

A la fin de la première partie, nous en étions restés à la navigation dans les différentes vues. Pour cela, il faut utiliser le sélecteur «Performance Warning», de la même manière qu'avec l'explorateur Windows. Voici quelques exemples d'utilisation de ce sélecteur :

Performance Warnings	
CPU Usage	les fonctions les plus coûteuses en termes d'utilisation CPU
Functions	
Performance Warnings	les fonctions les plus coûteuses intervenant dans la
Frames avec tri par CPU Time	la génération de la frame sélectionnée
Functions	
Performance Warnings	l'arbre visuel et le coût précis de chaque élément
Frames avec tri par CPU Time	(Measure/Arrange)
Visual Tree	

L'analyseur de performances peut vous indiquer où optimiser votre code. Vous pouvez ainsi concentrer vos efforts sur les méthodes les plus coûteuses. Bien souvent, deux options s'offrent à vous : optimiser le code en lui-même : charger moins d'éléments, optimiser un algorithme, différer une opération dans le temps... effectuer le travail dans un autre thread pour ne pas bloquer le thread UI. Afin de simplifier la gestion de l'asynchronisme dans votre application, vous pouvez utiliser les nouveaux mots clés *async/await* de la prochaine version du langage C#.

Profilage mémoire

Voici le type de résultat que l'on obtient si on choisit l'option «Memory» du profiler : [Fig.1].

La trace affiche l'utilisation de la mémoire au cours du temps, ainsi que des indicateurs. Lorsque l'on survole un indicateur, un tooltip



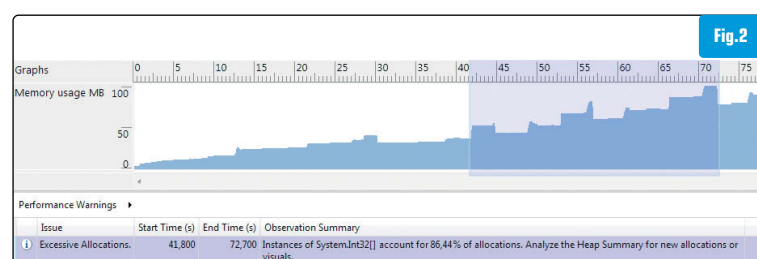
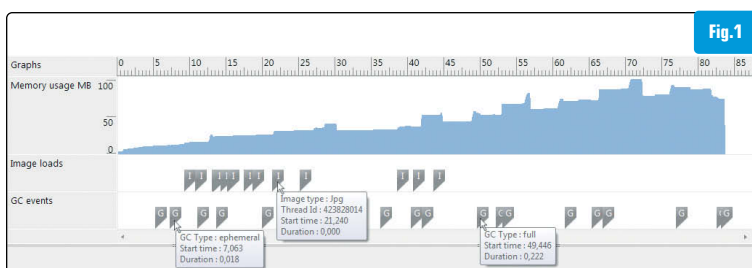
s'affiche avec plus de détails. Ces indicateurs marquent :

- les chargements des images, pouvant expliquer des pics d'utilisation mémoire
- les passages du garbage collector. Notez que le garbage collector est générationnel, avec deux niveaux de génération : Gen0 et Gen1. Le profiler distingue donc deux types de collecte : «ephemeral» et «full». Les collectes «ephemeral» sont plus rapides car elles ne concernent que les objets de génération 0.

Cette trace a été obtenue avec l'application 2Day, en éditant successivement 3 fois la même tâche. L'allure générale de la trace montre clairement un problème de fuite mémoire : la consommation mémoire augmente sans jamais redescendre. L'exception *OutOfMemoryException* n'est probablement pas très loin... L'outil va nous permettre de trouver la cause de cette fuite.

De la même manière que pour l'analyse de performances, vous pouvez sélectionner une partie du graphique pour vous concentrer sur une phase d'exécution donnée, et l'outil peut vous proposer des suggestions. Si nous sélectionnons la zone avec les 3 pics, nous obtenons une première indication : « Instances of Int32[] account for 86,44% of allocations ». Il semblerait que la grande majorité de la mémoire allouée corresponde à des tableaux d'entiers... [Fig.2].

Le menu « Heap Summary » permet d'inspecter les allocations qui



ont été faites sur le tas. La ligne « Retained allocations at end » indique les instances qui consomment de la mémoire. Voici les types concernés : [Fig.3].

Il n'y a plus vraiment de doute sur « qui » consomme la mémoire, reste à savoir d'où vient cette allocation. Il suffit de reprendre le sélecteur pour descendre au niveau des méthodes. On tombe ensuite sur le code (simplifié) suivant de 2Day :

```
private int[] data;
public EditTaskPage()
{
    InitializeComponent();
    this.workbook = Ioc.Resolve<IWorkbook>();
    this.data = new int[2500000];
    this.workbook.FolderAdded += this.OnFolderAdded;
}
```

Au-delà de l'allocation « brutale » d'un tableau de 2 500 000 entiers (qui était juste là pour mettre en évidence la fuite), nous découvrons aussi pourquoi la mémoire n'est pas relâchée : le constructeur de la page s'abonne à un événement sans jamais se désabonner. Comme l'objet source (workbook) a une durée de vie supérieure à celle de la page d'édition d'une tâche, on a créé une fuite mémoire !

Le « correctif » est ici assez simple : dans la méthode OnNavigatedFrom (appelée par le système quand on quitte la page), il suffit de se désabonner à l'événement « FolderAdded ». Notons que si nous avons utilisé une expression lambda pour l'abonnement, il n'aurait pas été possible de se désabonner !

L'outil d'analyse mémoire vous aide ainsi à vérifier que votre utilisation mémoire est correcte. De manière générale, un bon conseil est de faire très attention aux événements, et d'utiliser la mémoire avec parcimonie. Sachez que lorsque le garbage collector est déclenché, l'application est mise en pause. Il peut parfois être intéressant de forcer manuellement le passage du GC, aux moments les moins pénalisants pour l'utilisateur, comme par exemple au chargement d'une nouvelle page.

ASTUCES ET BONNES PRATIQUES POUR AMÉLIORER LES PERFORMANCES

> BitmapCache & Compositor

Revenons sur le nouveau thread spécifique à Windows Phone : le thread compositor. Ce thread prend en charge des animations « simples », afin qu'elles soient toujours fluides. Voici les quatre types d'animations supportées par le thread compositor :

- Translation (*TranslateTransform*)
- Perspective (*PerspectiveTransform*)
- Opacité (*OpacityTransform*)
- Clip rectangulaire

Le développement de l'application 2Day nous a donné l'occasion de vérifier cela.

Dans 2Day, afin que l'utilisateur ait à tout moment la liste de ses

Fig.3

Performance Warnings ▶ Heap Summary ▶ Types ▶			
Type Name	Instances	Total Size (Bytes)	Max Size (Bytes)
System.Int32[]	3685	30241000	10000012
System.String	4540	353340	44988

dossiers et l'indication du dossier couramment affiché, nous avons créé un contrôle (appelé PivotHeader) qui affiche la liste de tous les éléments du pivot, ainsi qu'un petit rectangle (appelé PivotIndicator, en rouge dans la capture ci-dessous) situé dessous l'élément actif. Lorsque la vue change (suite à une navigation vers un autre dossier), ce rectangle se déplace horizontalement pour indiquer le nouvel élément courant [Fig.4 et 5].

Pour réaliser cela, nous avons défini une animation basée sur la propriété Canvas.Left du rectangle :

```
Border rectangle = new Border();
DoubleAnimation animation = new DoubleAnimation
{
    Duration = new Duration(TimeSpan.FromSeconds(0.5)),
};
Storyboard.SetTarget(animation, rectangle);
Storyboard.SetTargetProperty(animation, new PropertyPath(Canvas.
LeftProperty));
```

Avec cette solution, l'animation n'était pas fluide. Nous avons alors utilisé l'option EnableCacheVisualization pour essayer de comprendre la source du problème :

```
Application.Current.Host.Settings.EnableCacheVisualization = true;
```

Cette option permet de visualiser en temps réel les parties de l'écran qui sont redessinées par le thread UI. Avec cette option activée, durant l'animation du PivotIndicator, nous observions que l'ensemble du PivotHeader clignotait, ce qui signifie qu'il était entièrement redessiné, alors que seule la position du rectangle devrait être mise à jour. La première amélioration apportée est l'activation du cache pour les éléments du PivotHeader (les icônes des dossiers) et le rectangle du PivotIndicator.

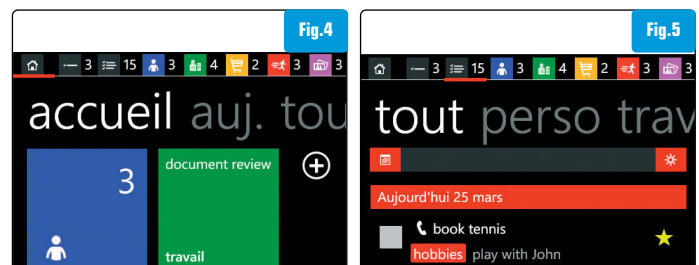
```
border.CacheMode = new BitmapCache();
```

Le principe de cette option est le suivant : les éléments visuels vont être stockés en cache sous forme de bitmaps après leur premier rendu, permettant de les afficher rapidement par la suite. Cette modification supprime le clignotement observé avec l'option EnableCacheVisualization, signe que les éléments ne sont plus redessinés inutilement ! Cependant, l'animation n'est toujours pas fluide.

Une autre option intéressante à utiliser est *EnableRedrawRegion*. Cette option permet de visualiser en temps réel les parties de l'écran qui sont accélérées graphiquement.

```
Application.Current.Host.Settings.EnableRedrawRegions = true;
```

Cette option met en évidence que le GPU ne gère pas l'animation du PivotIndicator. En effet, lorsque la propriété Canvas.Left change, cela cause une mise à jour du layout du Canvas qui va repositionner



ses éléments (phase dite de « Arrange »). Le GPU ne peut pas accélérer cette opération, c'est pourquoi le thread compositor ne gère pas ce type d'animation.

Nous avons donc remplacé notre animation initiale par un `TranslateTransform` supporté par le thread compositor, ce qui a résolu le problème de fluidité :

```
TranslateTransform translateTransform = new TranslateTransform();
Border rectangle = new Border
{
    RenderTransform = translateTransform
};
DoubleAnimation animation = new DoubleAnimation
{
    Duration = new Duration(TimeSpan.FromSeconds(0.5)),
};
Storyboard.SetTarget(animation, translateTransform);
Storyboard.SetTargetProperty(animation, new PropertyPath(TranslateTransform.XProperty));
```

Soyez donc vigilant quand vous définissez une animation, essayez au maximum d'utiliser des types pris en charge par le thread compositor !

> Visibility & Opacity

Selon le contexte, on a souvent besoin d'afficher ou de cacher dynamiquement des parties de l'IHM. Pour réaliser cela, vous pouvez jouer sur deux propriétés : `Visibility` et `Opacity`. Ce choix a un impact sur les performances à ne pas négliger, surtout lorsqu'il s'applique à un grand nombre d'éléments graphiques.

- La propriété `Visibility` peut prendre deux états en Silverlight : `Collapsed` et `Visible`. Si un élément graphique est dans l'état `Collapsed`, aucune donnée relative à cet élément n'est conservée en mémoire, aucun traitement graphique n'est réalisé pour cet élément. Mais lorsque l'élément repasse à l'état `Visible`, le thread UI doit entièrement le dessiner, ce qui nécessite des ressources et du temps.
- La valeur de la propriété `Opacity` est un double entre 0 et 1. Une opacité de 0 correspond à un élément complètement transparent : l'élément n'est pas visible, mais contrairement à l'état `Collapsed`, il est présent dans l'arbre visuel. L'opacité peut vous permettre d'améliorer les performances de votre application si vous l'utilisez conjointement avec l'option `BitmapCache` (`CacheMode = BitmapCache`). En effet, dans ce cas, lorsque l'opacité change, aucune phase de rendu n'est nécessaire pour réaliser la mise à jour de l'interface : le bitmap stocké en cache est simplement affiché.

> Media

Voici quelques bonnes pratiques pour gérer correctement vos images :

- Choisissez le bon format parmi les deux formats supportés : `jpg` et `png`. Le format `jpg` est à utiliser de préférence, car le décodage est plus rapide. Mais sachez qu'il gère moins bien les dégradés de couleurs. Si vous avez besoin de transparence, vous n'avez pas le choix et devez utiliser `png`.
- Grâce à l'option `CreateOptions=BackgroundCreation` de `BitmapImage`, déportez le décodage de vos images en arrière-plan, plutôt que de le laisser à la charge du thread UI.
- Pour vos éléments statiques, privilégiez l'utilisation d'une image plu-

tôt que du XAML. Le XAML nécessite plus de calcul, car il requiert une phase de parsing, puis de création et de rendu.

- Limitez la résolution des images à 2000 x 2000, taille maximale gérée par l'environnement Windows Phone. Si ce n'est pas possible, ne chargez qu'une partie de l'image, en utilisant un `WriteableBitmap` et la méthode d'extension `LoadJpeg`. Et de manière générale, créez directement vos images avec la bonne résolution utilisée dans l'application, afin de ne pas générer des calculs inutiles de redimensionnement.
- Etant donné que le traitement des médias n'est pas optimisé pour les flux mémoire, n'embarquez pas vos fichiers de média en tant que ressource, mais positionnez plutôt la propriété `BuildAction` à `Content` (le média sera un fichier dans le `.XAP` mais ne sera pas inclus dans l'assembly).

> Perception

Certaines opérations longues ne peuvent pas être « accélérées », par exemple dans le cas d'un échange avec un serveur à travers le réseau. Dans ces cas, les bonnes pratiques pour améliorer le ressenti de vos utilisateurs sont les suivantes :

- ne pas figer l'UI : effectuer l'opération en arrière-plan
- indiquer à l'utilisateur qu'une opération est en cours : utiliser par exemple le contrôle `PerformanceProgressBar`
- afficher l'avancement de l'opération, avec un message et un pourcentage

Dans `2Day`, nous avons mis en place ce principe avec un splashscreen animé. La plateforme permet déjà de personnaliser l'image de démarrage. Pour faire patienter l'utilisateur durant le lancement, nous avons ajouté de l'interactivité avec une première page contenant un message et une barre de progression affichée pendant le chargement de la page d'accueil.

CONCLUSION

Windows Phone 7 constitue une plateforme embarquée avec des ressources nécessairement plus limitées. En conséquence, l'optimisation des performances de vos applications est un sujet indispensable à ne pas négliger.

- Testez votre application en conditions réelles (sur un vrai device, avec une connexion 3G...) afin de déterminer si les performances sont suffisantes.
- Optimisez en permanence toutes les parties de votre application. Devenez un optimisateur fou en traquant la moindre ligne de code ☺.
- Sachez tirer parti de la plateforme (thread compositor...) pour améliorer vos applications.
- Utilisez les outils (compteurs, profiler et options de debug) : ils vous guident et vous permettent de savoir quelles parties du code optimiser.
- Votre devise : « Less Is More » ! Allégez au maximum votre application en ne créant les visuels qu'au dernier moment et à la demande, simplifiez votre arbre visuel...

Bonnes optimisations et à vous les ★★★★★ !

Charlotte Gaidon, charlotte.gaidon@gmail.com
Formatrice et développeur WPF/SL/WP7 - @nutchad

Jérémie Alles, jeremy.alles@gmail.com
MVP Client App Dev - www.japf.fr - @jalpf Ingénieurs pôle Microsoft chez THALES SERVICES à Grenoble

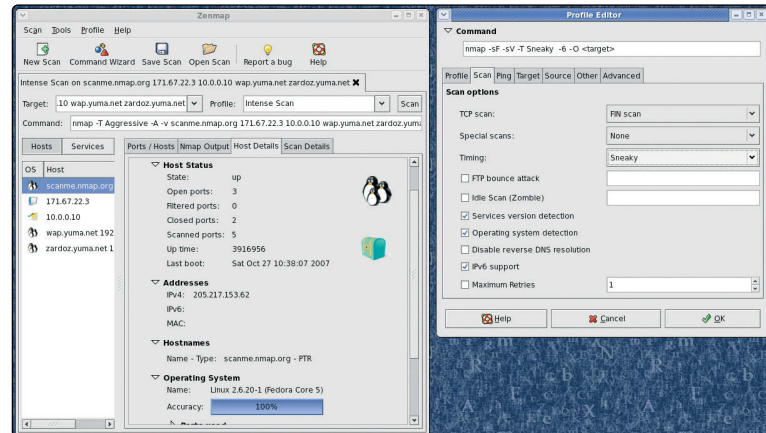
Scans de port : les comprendre, les utiliser, et s'en prémunir

Les scans de port, ou balayages de port, sont une technique pour déterminer quels services sont actifs sur une machine ou pour tester la résistance d'un pare-feu. A la limite de la légalité, l'administrateur les trouvera utiles pour connaître les failles de son système, mais il veillera aussi à s'en protéger. Prise de contact sous Linux.

Les scans, ou balayages, de port exploitent des particularités des protocoles réseaux en envoyant des paquets de données sur tous les ports d'une machine, d'où le nom de balayage, afin de faire des déductions. Ce port est-il ouvert, filtré, ou fermé ? Le port 110 est ouvert, quel est le service (qmail, postfix, ?...) qui tourne ? Ces déductions permettent aux hackers d'élaborer une stratégie pour leur attaque. Alors autant leur communiquer le moins d'informations possible et leur fermer la porte au nez ! Les scans de ports sont un outil puissant qui permet même parfois de déterminer un service actif sur une machine pourtant protégée par un pare-feu. En outre les scans de port, souvent utilisés par des script kiddies, c'est-à-dire des pirates en herbe qui utilisent les outils créés par des experts, peuvent aboutir à une utilisation excessive de la pile réseau d'une machine attaquée, qui peut se retrouver en situation de déni de service. Là encore, mieux vaut arrêter l'hémorragie dès que possible. Les scans de port ont aussi leur utilité pour l'administrateur. En scannant une machine dont on vient de vous confier l'administration, vous pouvez avoir bien des surprises, et découvrir que des services vulnérables sont ouverts aux quatre vents. Vous pouvez aussi découvrir la présence d'un service totalement inconnu de vous. Qui pourrait bien être une backdoor installée par un pirate ! Le cas a été très fréquemment rencontré. N'hésitez donc pas à vous scanner vous-même régulièrement. C'est même la seule façon d'utiliser le scan de port qui soit de bon aloi.

1 L'ASPECT LÉGAL

Les articles 323 1 à 323 7 du Code Pénal sur les atteintes aux systèmes de traitement automatisé de données considèrent que la prise d'empreintes de ports informatiques ne peut pas être assimilée à un accès frauduleux à un système de traitement automatisé de données. Cependant il peut être considéré qu'une telle action s'inscrit dans une démarche visant à nuire à un système. C'est alors au tribunal d'apprécier la question et éventuellement de sanctionner. Pour un administrateur système, paranoïaque par définition, un scan de port est une attaque ! Il se plaindra donc à votre hébergeur, qui se retournera sans doute contre vous. A ce sujet, certains hébergeurs sont très chatouilleux sur la question. Ainsi si vous avez deux serveurs chez un même hébergeur, ne croyez pas que vous pouvez scanner l'un avec l'autre. L'hébergeur va détecter cette activité sur son réseau et peut fort bien mettre hors service la machine qui a lancé le scan. Si vous voulez scanner une machine avec une autre, vous devez avant toute chose prévenir votre hébergeur, qui vous laissera alors faire.



2 NOTIONS DE PROTOCOLE RÉSEAU

Sur un réseau IP, ce qu'est Internet, deux protocoles sont principalement utilisés : UDP et TCP. Dans les deux cas, la communication se fait par envoi de donnée dans un paquet constitué d'un en-tête et desdites données. Intéressons-nous plus particulièrement au protocole TCP. Dans les protocoles, les paquets sont constitués ainsi :

0	15	31

port source	port destination	

numéro de séquence		

numéro d'accusé réception		

HL	rsvd	taille de fenêtre

somme de contrôle TCP	pointeur d'urgence	

Ce qui retiendra toute notre attention ici sont les bits de contrôle que les systèmes s'échangent, lorsqu'une connexion s'établit, se termine ou est réinitialisée, ainsi que les numéros de séquence. Lorsqu'un client se connecte à un serveur pour établir une connexion, il envoie un paquet avec un numéro de séquence et le bit SYN (S) positionné. Le serveur lui répond en retournant un paquet dont les bits SYN-ACK (S-A) sont positionnés, le numéro d'accusé réception (ACKnowledgement) est le numéro de séquence reçu et incrémenté. Recevant cela, le client retourne un paquet avec le bit ACK positionné, et ainsi de suite. Le processus de scan de port, lui, enverra des paquets mal constitués du point de vue du réseau, mais savamment

bidouillés :) Vous pouvez observer le déroulement d'une connexion TCP normale avec l'outil TCPDUMP. Par exemple pour une machine faisant office de serveur, avec Apache qui y tourne et connectée à votre réseau local avec une interface WIFI, vous pouvez donner dans un terminal :

```
#sudo tcpdump -n -i wlan0 port http

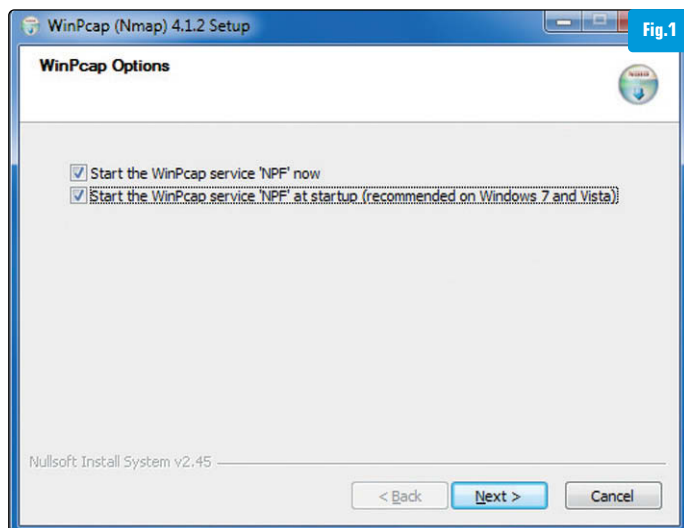
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wlan0, link-type EN10MB (Ethernet), capture size 65535 bytes
```

Si vous avez une connexion filaire, vous mettez eth0 à la place de wlan0. Puis, depuis une autre machine de votre réseau, sollicitez le serveur, par exemple en chargeant la page d'accueil :

```
wget http://192.168.1.10
```

Vous obtenez :

```
IP 192.168.1.12.20225 > 192.168.1.10.80: Flags [S], seq 572605798, win 8192, options [mss 1460,nop,nop,sackOK], length 0
IP 192.168.1.10.80 > 192.168.1.12.20225: Flags [S.], seq 3177261357, ack 572605799, win 14600, options [mss 1460,nop,nop,sackOK], length 0
IP 192.168.1.12.20225 > 192.168.1.10.80: Flags [.], ack 1, win 64240, length 0
IP 192.168.1.12.20225 > 192.168.1.10.80: Flags [P.], seq 1:111, ack 1, win 64240, length 110
IP 192.168.1.10.80 > 192.168.1.12.20225: Flags [.], ack 111, win 14600, length 0
IP 192.168.1.10.80 > 192.168.1.12.20225: Flags [P.], seq 1:492, ack 111, win 14600, length 491
IP 192.168.1.12.20225 > 192.168.1.10.80: Flags [F.], seq 111, ack 492, win 63749, length 0
IP 192.168.1.10.80 > 192.168.1.12.20225: Flags [F.], seq 492, ack 112, win 14600, length 0
IP 192.168.1.12.20225 > 192.168.1.10.80: Flags [.], ack 493, win 63749, length 0
```



Dups ! MySQL ne devrait pas être vu ici. Comme quoi les scans de port peuvent avoir du bon.

Au début de cette sortie de tcpdump, on voit bien la communication avec les bits Syn et Ack. Le reste de la sortie est intéressante en ce qui concerne les numéros d'ordre, mais ceci sort du champ de cet article.

4 NMAP

Il existe de nombreux outils de scan de port, mais le numéro un est incontestablement Nmap. C'est un utilitaire Linux, mais il en existe une version Windows qui fonctionne fort bien, pourvu que vous installiez aussi WinPcap (qui vient avec la distribution de Nmap), et que le service WinPcap soit démarré [Fig.1]. Dès Nmap installé, lancez-le pour un scan rapide, de base, sur un de vos serveurs ou le serveur d'un de vos amis, que vous aurez bien sûr prévenu auparavant :

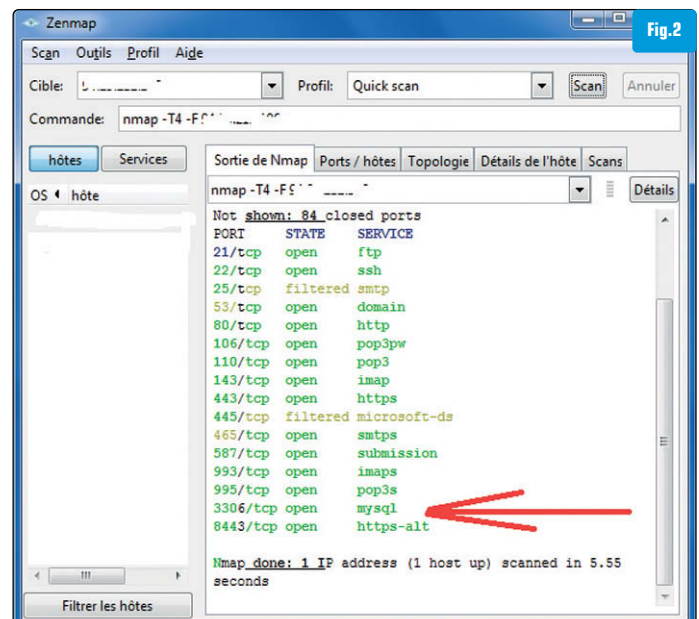
```
nmap -T4 -F 192.168.1.10
```

Cette simple action peut vous mettre en évidence une étourderie. Ainsi comme le montre l'illustration, [Fig.2], le scan du serveur d'une de mes connaissances a montré que le service MySQL y était accessible depuis tout Internet, ce qui vraiment ne s'impose pas.

5 QUELQUES TECHNIQUES DE SCANS

Nmap est un outil extrêmement puissant, et même précieux, pour approfondir ses connaissances. Il permet d'effectuer les scans les plus avancés. Le lecteur intéressé aura tout intérêt à étudier attentivement sa documentation, puis à réfléchir sur les conclusions que l'on peut tirer en croisant les informations obtenues au moyen de plusieurs scans habiles. En voici quelques-uns, décrits succinctement :

- **sS** : Cette option de Nmap lance un scan SYN, que l'on appelle encore scan semi-ouvert. Cela consiste en l'envoi d'un paquet avec le bit SYN positionné, comme pour l'établissement d'une connexion. On attend la réponse du serveur, mais on n'y répond pas pour finir d'établir la connexion, d'où le nom de semi-ouvert. Si le serveur répond par un paquet avec les bits SYN-ACK, cela signi-



Pour profiter au mieux de Nmap sous Windows, lancez le service WinPcap.

fié qu'il suit le protocole, donc que le port est ouvert. S'il répond avec un bit RST (RESET), cela signifie que le port est fermé. Et si après plusieurs essais le serveur ne répond pas ? C'est que le port est filtré, ce qui peut vouloir dire que l'on touche un pare-feu, ou un routeur.

- sA : Vous supposez avoir à faire à un pare-feu, vous voulez vous en assurer ? Essayez le scan ACK. Le principe consiste à envoyer des paquets avec le bit ACK positionné. Sans passer d'abord par la phase SYN du protocole. Ce scan ne vous dira pas si un port est ouvert. Mais si un bit RST est reçu en retour, cela veut dire qu'un noyau n'est pas content que le protocole ne soit pas respecté (absence du bit SYN). Le noyau brise donc là la communication. S'il n'y a pas de réponse c'est un pare-feu :-)
- sX, dit scan Xmas, Christmas ou arbre de Noël. Il envoie les bits FIN, PSH et URG. Sa particularité est de traverser certains pare-feu ou routeurs sans état de connexion.

6 LES SCANS TRÈS AVANCÉS

Vous devez, au moyen de Nmap, les essayer au moins une fois sur les serveurs dont vous avez la responsabilité. Car si vos serveurs y sont vulnérables, c'est vous qui pouvez être accusé de scan. En effet le scan dit Zombie (-sl) est une technique étonnante, qui consiste à faire effectuer le scan non par la machine de l'attaquant, mais par une machine tierce dite zombie.

Il existe encore le scan par rebond FTP qui consiste à exploiter une particularité du protocole FTP. Par cette technique, un attaquant peut faire scanner une machine via votre serveur FTP, si celui-ci est vulnérable. Vous l'avez compris, se protéger des scans de port est plutôt une bonne idée. Il existe plusieurs outils pour cela. Je vous propose de découvrir Portsentry.

7 SE PROTÉGER DES SCANS DE PORT AVEC PORTSENTRY

Portsentry est un outil qui surveille les ports non utilisés normalement (80, 21, etc...) par votre système. Quand un balayage de port touche les ports surveillés par Portsentry, celui-ci écrit dans le fichier `/var/log/syslog`, et, selon sa configuration, bannit en plus l'hôte émetteur du balayage, configure le pare-feu à la volée, bloque une route sur le réseau, ou exécute une commande définie par l'administrateur. Portsentry présente des points communs avec fail2ban étudié dans Programmez 153 (juin 2012). Et comme avec lui, vous devez veiller à ne pas vous bannir vous-même définitivement d'un serveur lors d'essais. Vous seriez contraints de redémarrer votre machine, perspective absolument insupportable pour tout administrateur qui se respecte :-). Bien noter que si fail2ban configure le pare-feu pour un temps déterminé, les actions de Portsentry sont en revanche définitives ! Portsentry s'installera très simplement selon votre distribution. Pour cet article nous avons utilisé une Ubuntu, basée Debian comme chacun sait. Donc :

```
apt-get install portsentry
```

Après installation, Portsentry peut-être lancée tout simplement comme une application, mais on préférera la lancer comme un démon :

```
sudo service portsentry start
```

Ou peut-être, selon votre distribution

```
/etc/init.d/portsentry start
```

Portsentry vient avec plusieurs fichiers de configuration. Si vous avez une IP fixe, mettez là dans `/etc/portsentry/portsentry.ignore.static`. Ce faisant vous vous assurez de ne pas vous bannir vous-même. Nous verrons plus loin une solution pour les IP dynamiques. Visitez ensuite le fichier `/etc/portsentry/portsentry.conf`. Vous y voyez des listes de ports à surveiller pour chacun des protocoles TCP et UDP.

```
# Use these if you just want to be aware:
TCP_PORTS="1,11,15,79,111,119,143, # etc..."
UDP_PORTS="1,7,9,69,161,162,513,635 # etc"
```

Bien sûr dans cette liste n'apparaissent pas les ports de vos services normalement actifs, comme par exemple 22 et 80. Quand portsentry démarre, il écoute les ports listés ici. Et bien entendu, il ne peut écouter un port qui est déjà ouvert normalement par un service. Vous pouvez encore définir quels ports feront l'objet d'une surveillance avancée :

```
ADVANCED_PORTS_TCP="1024"
ADVANCED_PORTS_UDP="1024"
```

Ici, seulement les ports en dessous de 1024, ou ports administrateurs feront l'objet d'une surveillance avancée. Que doit faire Portsentry lorsqu'il découvre une attaque ? Vous le spécifier ici :

```
# 0 = Do not block UDP/TCP scans.
# 1 = Block UDP/TCP scans.
# 2 = Run external command only (KILL_RUN_CMD)

BLOCK_UDP="2"
BLOCK_TCP="2"
```

Par défaut c'est la valeur 0 qui est sélectionnée. Dans ce cas Portsentry ne fait rien d'autre que de consigner les attaques qu'il a détectées dans le fichier `/var/log/syslog`. Bien faire attention à l'option 1. Avec cette option, Portsentry bannit l'attaquant en ajoutant une ligne dans le fichier `/etc/hosts.deny`. C'est idéal pour vous bannir vous-même ! C'est pourquoi je vous déconseille non seulement d'utiliser cette option, mais de mettre en plus une ligne en commentaire dans `/etc/portsentry.conf` :

```
#KILL_HOSTS_DENY="ALL: $TARGET$ : DENY"
```

L'option 2 est indéniablement la plus intéressante car elle vous permet de définir finement l'action effectuée par Portsentry. Pour cela vous donnez une commande, ou même un script Shell. Vous pouvez par exemple écrire :

```
KILL_RUN_CMD="/sbin/iptables -A INPUT -s $TARGET$ -j DROP"
```

`$TARGET$` sera remplacé par l'adresse IP de l'attaquant. Avec cette commande tout ce qui provient de cette adresse sera rejeté par le pare-feu. Si cette adresse est la vôtre, vous vous êtes encore banni irrémédiablement. Une solution pour éviter cela est d'écrire :

```
KILL_RUN_CMD="/sbin/iptables -A INPUT -s $TARGET$ -p tcp --dport 22 -j ACCEPT && /sbin/iptables -A INPUT -s $TARGET$ -j DROP"
```

Attention à l'ordre des commandes iptables qui *n'est pas indifférent* ! Avec cette commande, vous ouvrez d'abord le pare-feu pour les connexions sur le port 22 en provenance de l'adresse \$TARGET\$ puis vous refusez tout le reste. De cette façon, il vous sera toujours possible d'ouvrir un terminal en SSH pour intervenir sur votre système :) Essayez quand même une première fois sur deux machines en réseau à la maison ;) Ce n'est pas un inconvénient de laisser ici ouvert le port 22. D'ailleurs ce port est ouvert normalement. En revanche, deux précautions valant toujours mieux qu'une, il est très pertinent de protéger en plus votre port 22 au moyen de fail2ban, comme nous l'avons vu dans Programmez! 153. Il est pertinent de ne pas se limiter à fermer le pare-feu en cas d'attaque. Un bon administrateur système aime savoir ce qui se passe dans ses machines en temps réel, plutôt que consulter les logs de temps à autre. Ainsi, vous pouvez préférer faire exécuter par KILL_RUN_CMD un script Shell qui commencera par vous envoyer un mail contenant un extrait de log, pour vous informer de ce qui se passe. L'écriture de ce script est laissée à la sagacité du lecteur. Il y a encore un fichier de configuration qui mérite votre attention : /etc/default/portsentry. Vous y trouvez :

```
TCP_MODE="tcp"
UDP_MODE="udp"
```

Ce qui signifie que Portsentry fait une détection simple par défaut. Si avec cette configuration de base vous faites un essai :

```
nmap -sS -v 192.168.1.10
```

Vous aurez la surprise de trouver un /var/log/syslog désespérément vide. Nous avons effectué un scan furtif qui n'est pas détecté. Donc n'hésitez pas, configurez ainsi :

```
TCP_MODE="stcp"
UDP_MODE="sudp"
```

pour avoir une détection avancée, ou si vous voulez vous limiter pour cela au port administrateur, comme expliqué plus haut, donnez :

```
TCP_MODE="atcp"
UDP_MODE="audp"
```

Bien entendu, tout changement de configuration, dans quelque fichier que ce soit ne prend effet qu'après le redémarrage du démon de Portsentry :

```
service portsentry restart
```

Après quoi vous pourrez voir Portsentry en action dans /var/log/syslog :

```
nmap -sS -v 192.168.1.10
```

```
portsentry[4351]: attackalert: TCP SYN/Normal scan from host:
192.168.1.12/192.168.1.12 to TCP port: 143
portsentry[4351]: attackalert: Ignoring TCP response per
configuration file setting.
portsentry[4351]: attackalert: TCP SYN/Normal scan from host:
192.168.1.12/192.168.1.12 to TCP port: 111
portsentry[4351]: attackalert: Host: 192.168.1.12/192.168.1.12
is already blocked Ignoring
# etc, etc.
```

ou encore

```
nmap -sX -v 192.168.1.10
```

```
portsentry[4351]: attackalert: TCP XMAS scan from host:
192.168.1.12/192.168.1.12 to TCP port: 143
portsentry[4351]: attackalert: Host: 192.168.1.12/192.168.1.12
is already blocked Ignoring
portsentry[4351]: attackalert: TCP XMAS scan from host:
192.168.1.12/192.168.1.12 to TCP port: 111
portsentry[4351]: attackalert: Host: 192.168.1.12/192.168.1.12
is already blocked Ignoring
```

Frédéric Mazué - fmazue@programmez.com



**ABONNEMENT
PDF**
30 € par an
**soit 2,73 € le
numéro**

www.programmez.com

**Abonnement
INTÉGRAL**

Pour un supplément de 10 € an
**accès illimité
aux archives**

Cette option est réservée aux abonnés pour 1 an au magazine, quel que soit le type d'abonnement (Standard, Numérique, Etudiant). Le prix de leur abonnement normal est majoré de 10 € (prix identique pour toutes zones géographiques). Pendant la durée de leur abonnement, ils ont ainsi accès, en supplément, à tous les anciens numéros et articles/ dossiers parus.

Chronique d'un bug dû à l'utilisation des délégués en C#

N.B. : S'applique à tout .NET (WPF, Silverlight, ASP.NET, Windows 8 XAML+C#, ...)

Les applications aujourd'hui doivent tenir compte de l'accès sporadique au réseau. Intervenant sur le développement d'une application pour tablette connectée en 3G, et afin d'améliorer l'expérience pour l'utilisateur, j'ai naturellement décidé de mettre les appels aux services Web en file d'attente afin qu'ils soient effectués en arrière-plan par un thread de travail. Des appels asynchrones, autrement dit. Rien de plus simple grâce aux délégués : j'ajoute une classe avec le code suivant : voir code complet sur www.programmez.com

Le principe est simple : on conserve dans une collection les méthodes à appeler sous forme de délégués, et il n'y a plus qu'à les invoquer via un thread de travail. Ainsi, le code qui était :

```
Service.SapTransfer(SelectedProcess.SAPName);
```

Devient :

```
OutboundHelper.Enqueue(() =>
    Service.SapTransfer(SelectedProcess.SAPName)
);
```

Note : dans le code ci-dessus, *SAPName* est de type *System.String*.

A priori, cela devrait marcher. Sauf qu'un test unitaire m'alerte immédiatement : les appels du service ne sont pas faits correctement. Le test en question demande l'appel de la même méthode de service deux fois avec des paramètres différents. Au début, je crois que c'est le test unitaire qui ne fonctionne plus, car il indique que l'appel est effectué deux fois avec les paramètres, et surtout ce peut être au hasard les paramètres du premier ou du second appel qui sont répétés. Pour simplifier, alors que j'appelle f(A) puis f(B), le test me dit que j'ai eu f(A) suivi de f(A) ou bien f(B) suivi de f(B). Et de manière aléatoire : de quoi devenir fou !

CE QUI SE PASSE

Le fait d'obtenir des résultats différents de manière aléatoire alors que l'appel semble déterministe fait penser à des problèmes de multithreading, surtout qu'ici il y a bien un thread de travail. Mais le problème vient de l'utilisation de délégués pour l'appel différé.

En effet, quand j'écris le code suivant, c'est la référence vers *SelectedProcess.SAPName* qui est conservée avec le délégué, autrement dit l'adresse mémoire et pas la valeur :

```
OutboundHelper.Enqueue(() =>
    Service.SapTransfer(SelectedProcess.SAPName)
);
```

Or, quand le délégué sera récupéré, il est fort probable que *SelectedProcess.SAPName* ait une autre valeur. Et c'est ce qui se passe dans mon cas. Ainsi, si j'appelle :

```
SelectedProcess.SAPName = "A";
OutboundHelper.Enqueue(() =>
    Service.SapTransfer(SelectedProcess.SAPName)
);
SelectedProcess.SAPName = "B";
```

```
OutboundHelper.Enqueue(() =>
    Service.SapTransfer(SelectedProcess.SAPName)
);
ProcessWaitingActions();
```

L'appel effectif sera équivalent à :

```
Service.SapTransfer("B");
Service.SapTransfer("B");
```

Et cela vaut pour tous les paramètres de type référence passés à un délégué. Car c'est la référence qui est passée, alors même qu'on n'utilise pas le mot clé « ref ». Pour les paramètres de type valeur (comme *System.Int32* ou les structures), pas de problème car il sont bien passés par copie.

COMMENT L'ÉVITER

Bien sûr, pour éviter ce genre de problème, il faut prendre soin de passer une copie des paramètres, au lieu de les passer par référence. Dans mon cas, il faut écrire :

```
string sapName = SelectedProcess.SAPName;
OutboundHelper.Enqueue(() =>
    Service.SapTransfer(sapName)
);
Au lieu de :
OutboundHelper.Enqueue(() =>
    Service.SapTransfer(SelectedProcess.SAPName)
);
```

Mais je dois dire que même en le sachant, on tombe dans le piège de nouveau. D'où l'importance cruciale des tests unitaires, qui garantissent que les sorties sont bien ce que l'on attend pour des entrées données.

Le propre des problèmes à apparition aléatoire, comme dans le cas du multithreading ou notre cas de délégués, c'est qu'ils sont difficiles à détecter car deux exécutions similaires n'auront pas forcément le même résultat. Dans mon cas, le bug ne pouvait pas apparaître tant qu'on n'ajoutait pas deux fois la même méthode dans la liste, autrement dit il serait passé inaperçu dans la plupart des tests fonctionnels. Sans outillage adapté, on se retrouve à découvrir les bugs en phase de tests d'intégration (autrement dit, très tard) et à galérer pour les reproduire car en général le simple fait de placer un point d'arrêt modifie fondamentalement l'exécution au point de ne plus avoir le même résultat. L'outillage adapté, c'est notamment le fait d'avoir des tests unitaires automatisés d'assez haut niveau, pour intercepter ce genre de comportement non prévu. Ce sont eux qui m'ont sauvé, et je ne peux que vous encourager à en avoir suffisamment pour vous épargner de lourdes recherches de bugs.

Arnaud Weil

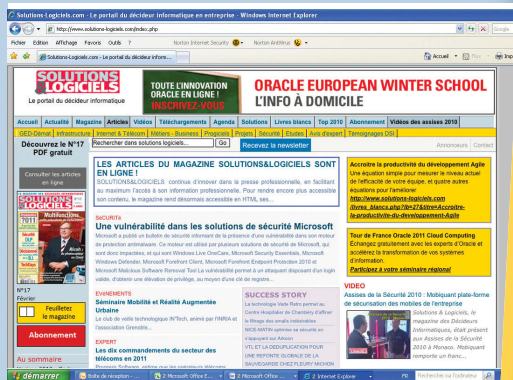
Développeur et formateur freelance

Site : <http://aweil.fr>

Les outils des Décideurs Informatiques

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

Cas clients
Actu triée par secteur
Avis d'Experts



Actus / Evénements / Newsletter / Vidéos



www.solutions-logiciels.com

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

Envoyer par la poste à : Solutions Logiciels, service Diffusion, GLIE - 17 chemin des Boulangers 78926 Yvelines cedex 9 - ou par fax : 01 55 56 70 20
1 an : 50€ au lieu de 60€, prix au numéro (Tarif France métropolitaine) - Autres destinations : CEE et Suisse : 60€ - Algérie, Maroc, Tunisie : 65€ , Canada : 80€ - Dom : 75€ Tom : 100€
10 numéros par an.

☐ M. ☐ Mme ☐ Mlle Société

Titre : Fonction : ☐ Directeur informatique ☐ Responsable informatique ☐ Chef de projet ☐ Admin ☐ Autre

NOM Prénom

N° rue

Complément

Code postal : | | | | | Ville

Adresse mail

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

A large, dark grey elephant sculpture is the central focus of the advertisement. It is positioned in a modern office environment with a high ceiling, large windows, and concrete pillars. The elephant is standing on its trunk, which is balanced on a single white tusk. Its body is arched, and its legs are spread out in a dynamic pose. The floor is made of large, light-colored square tiles that reflect the elephant and the office lights. The overall atmosphere is one of modernity and innovation.

TELECHARGEZ GRATUITEMENT !
Enterprise Developer Personal Edition,
Un environnement intégré et puissant pour le
développement de vos applications mainframe.
www.microfocus.com/EDPE

Découvrez l'intégralité de la gamme Enterprise
online.microfocus.com/awareness-fr

Transformer l'éléphant dans la pièce