

NOUVELLE FORMULE

Décembre 2000 • N°27 - 39F

LE 1^{ER} MAGAZINE DE TOUS LES LANGAGES

Programmez!

Programmez!

Ce mois-ci : XML WML ASP OPENGL JAVA C++ APACHE PYTHON

Sécuriser Windows

NT-2000

Les armes du cryptage X509-RC6

COCOON

La nouvelle infrastructure
de publication pour le Web

Des vitamines C++ dans Java

Sur le CD-Rom p.35

Nouvelle version JBUILDER 4.0

(Pour Windows, Linux et Solaris)

+ Microsoft .NET SDK

BELGIQUE 260 F.B. - SUISSE 12 F.S.
LUXEMBOURG 260 F.LUX - CANADA 8,95 \$ CAN

T 4319 - 27 - 39,00 F



Pratique

- ASP + par l'exemple
- Maîtriser Python en un éclair
- Premières réalisations OpenGL
- Débuter en Wap



Sommaire

Société Européenne de Presse
et de Communication
5-7 rue de l'Amiral-Courbet,
94160 Saint-Mandé - France
Tél. : 01 43 98 22 22 - Télécopie : 01 43 98 72 12

Éditeur délégué :
Stéphane Larcher

Rédaction
Directeur des rédactions : Michel Barreau
Rédacteur en chef : Stéphane Larcher
Secrétaire de rédaction : François Denivet
(francois.denivet@sepc.com.fr)
Assistante de rédaction : Apie-Josa Akaffou
(apie.josa@programmez.com)

Ont collaboré à ce numéro...

Bryan Aker, Cheng Chai Ang, Christophe Babayou,
Patrick Chambet, Richard Clark, Arnaud de la
Pommeraye, Alexandre Deloy, Jacques de Schryver,
Laurent Foynard, Xavier Leclercq, Denis le Guilf,
Frédéric Mazué, Gilles Mergoil, Carole Pitras, Jean-
Marc Quéré

Maquette

Chef de studio : Marc Soria-Piles
Première maquettiste : Véronique Belpois,
Maquettistes : Claude Marrel, Laurent Pleissingier

Services en ligne (ECC Multimédia)
Directeur : Stéphane Kaminsky
(sk@ecc-multimedia.net)

Multimédia/CD-Rom

Directeur : Olivier Delacourt (381)
(olivier.delacourt@sepc.com.fr)

Contenu logiciels :

René Robelin et Franck Cayrol
(hotline-cdrom@sepc.com.fr)

Intégration et développement :

Gilles Mélinat et Philippe Coupez

Publicité

Directeur commercial : Philippe Demyttenaere (43 86)

Directeurs de clientèle : Bernard Bibas (43 65)

et Virginie Leconte (88 36)

Chefs de publicité : Arnaud Fourest (43 60)

et Christophe Benoiste (43 61)

Assistant marketing : Benoît Gagnaire

Assistante de publicité : Christine Gaugry (43 83)

Diffusion/promotion

Directeur de la diffusion : Thierry Cagnion

Responsable abonnements et promotion :

Vanessa Pénélaud

vanessa.penelaud@sepc.com.fr

Sepecom Service Abonnements

22, rue René Boulanger, 75482 Paris Cedex 10

Tél. : 01 55 70 70 55 - Fax : 01 55 56 70 20

du lundi au jeudi

de 9h30 à 12h30 et de 13h30 à 17h00

le vendredi de 9h00 à 12h00 et de 14h00 à 16h30

abonnements.sepc.com@groupe-gli.com

Fabrication

Daniel Nardon

Directeur de la publication,

Président-directeur général

Jean Kaminsky

Groupe Sepecom

SA au capital de 450 000 F

Impression

Imprimerie RotoFrance

Numéro de commission paritaire :

0900 K 78366

Dépôt légal : 4^e trimestre 2000

Certains articles signalés sont copyright Linux

Journal © publiés sous contrat de licence

Ce numéro comporte sur la totalité des exemplaires 2 encarts

abonnement brochés entre les pages 34 et 35, et les pages

66 et 67, un encart abonnement jeté et un CD-Rom.

Tarifs d'abonnement à *Programmez !*

Abonnement (magazine seul)

1 an - 11 numéros

France métropolitaine : 299 FF TTC

CEE et Suisse : 340 FF TTC

Algérie, Maroc, Tunisie : 367 FF TTC

Canada : 422 FF TTC

Autres pays : nous consulter

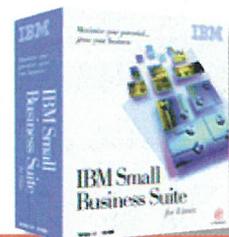


HOT LINE CD-ROM

par Minitel : 3614 ISICOM
par Internet : cdrom@sepc.com.fr
par téléphone : 01 43 98 43 89
du lundi au vendredi

ACTUALITÉS

Linux 2.4 pour Noël	6
En avant pour le V92	8
Version 4 de Dreamweaver et Fireworks	10
Livres, notre sélection	12



TEST

JBuilder 4.0, la version café très fort	14
---	----

PRATIQUE

Échanger des données en XML avec Notes et Domino	20
La Bourse au cours du Wap	30
La JNI unit C à Java	36
Python – Commandez votre déjeuner via le Web	44
Initiation à Open Gl – 2 ^e partie	48
Création de pagelets avec ASP+	54



REPORTAGE

Comment SDV met les journaux en ligne	58
---------------------------------------	----

TECHNOLOGIE

Apache – Cocoon sort de sa chrysalide	62
Voici ModLayout. Ça colle?	66



DOSSIER SÉCURITÉ

Ouverture	75
Windows NT/2000 – Attention, zone protégée	76
RC6 – le dernier algorithme de chiffrement de RSA	82
X509, la mise en œuvre des certificats d'authentification	86



TRAJECTOIRE

Laurent Ribardière, la passion du code	90
Histoire des langages – Que Forth soit avec toi	92
Programmation cocooning : le magnétoscope	94
Diététique – Sortir du régime pizza-cola	96
Recrutement	98





Programmez 2.0!

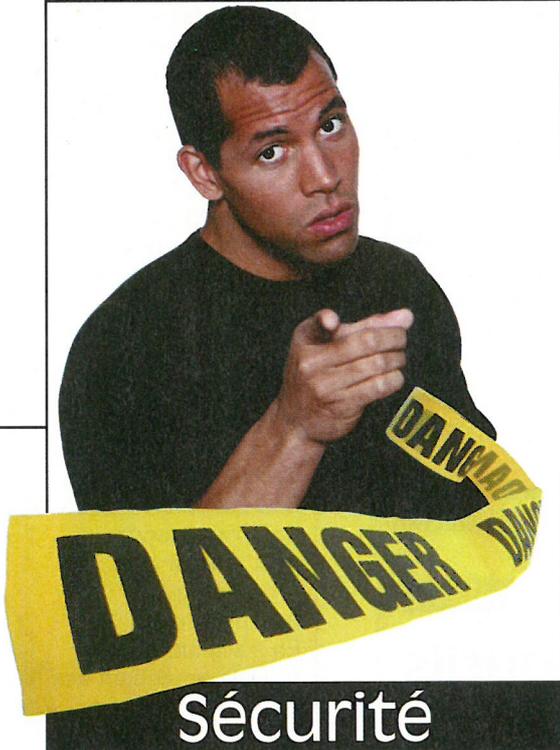


Stéphane Larcher
stephane.larcher@sepcom.fr

Lorsqu'il y a quelques mois, nous avons évoqué l'idée d'une nouvelle formule de *Programmez!*, les idées ont fusé de

partout comme à l'habitude dans ces circonstances. Et si on rajoutait ceci ? Et si on supprimait cela ? Et si on changeait le titre ? Et si, et si ? Finalement, il a bien fallu trancher et nous avons décidé de ne rien supprimer de ce qui existait déjà. Nous avons modifié la maquette pour la rendre – nous l'espérons – plus claire et plus facile à lire, comme on améliore une interface. Nous avons réorganisé les rubriques et en avons ajouté quelques-unes comme on le fait avec les fonctionnalités. C'est ainsi que vous retrouverez votre journal enrichi de reportages, de portraits et de quelques articles plus « décontractants ». Pour le reste, nous souhaitons rester le magazine de tous les langages et de tous les programmeurs comme vous le découvrirez dans les pages Tests, Pratique et Technologie. N'hésitez pas à nous faire part de vos points de vue sur ce que vous aimez ou regrettez dans cette version 2.0 et ce afin que la prochaine version soit toujours plus complète et toujours plus conforme à vos attentes.

EN COUVERTURE



Sécurité

> *On vous épie ? Vous avez des angoisses sécuritaires ? Vous craignez les « mots d'amour » ? Nous allons vous aider à protéger vos données contre les intrusions et attaques de toutes sortes.*

OS

Le noyau Linux 2.4 annoncé pour décembre 2000

Linus Torvalds vient de l'annoncer, la première version stable officielle de la nouvelle génération du noyau Linux verra le jour début décembre. Linus reconnaît que le kernel souffre encore de quelques bogues mineurs à corriger. Mais tout devrait être réglé dans les quelques semaines qui nous séparent de la date de sortie. Le noyau 2.4

est attendu comme l'entrée définitive de Linux dans sa phase de pleine maturité. Certaines nouveautés d'importance seront introduites comme un meilleur support de l'USB, et une couche réseau totalement refaite. Nombre de distributeurs commerciaux (RedHat, Suse, Mandrake ou encore Caldera) se disent d'ores-et-déjà prêts pour distribuer des

paquetages adaptés à leurs produits respectifs du nouveau noyau.

Linus ne cache pas que cette date de sortie lui convient également pour des raisons d'ordre personnel. La famille Torvalds attend un troisième enfant qui devrait naître dans la même période. On comprend que l'heureux papa aimerait conclure cette nouvelle version du noyau avant. ●

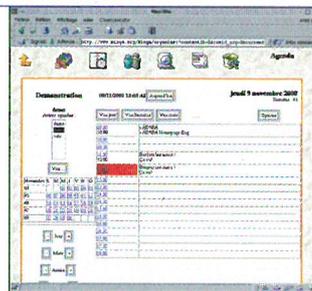
Groupware

Le « prêt à partager », nouvelle ligne d'Atrid

Atrid, SSII spécialisée dans les solutions basées sur les logiciels libres, vient de placer en libre téléchargement MIOGA, son logiciel de « groupware ». Comme tout logiciel libre, MIOGA utilise les standards ouverts tels que le XML et le client fonctionne à partir d'un simple navigateur web.

L'archive tarball proposé sur le site de la société est numérotée 0.18 et ne dispose pour l'instant que d'une documentation très spartiate. MIOGA nécessite un serveur Apache, et la base de données PostgreSQL. Vous devez également disposer de différents modules

Perl. Ce système propose toutes les fonctions nécessaires au travail de groupe : agendas partagés, gestion des tâches et des plannings, système de dossiers communs, etc... Bref, un logiciel sous licence GPL qui pourrait donner aux PME une nouvelle bonne raison de préférer Linux pour tous les serveurs d'applications et de services. ●



<http://www.mioqa.com/>
<http://www.atrid.com/>

E-commerce

IBM lance une suite d'outils Linux destinée aux PME



IBM Small Business Suite est un ensemble de programmes visant à centraliser les services informatiques des PME et à les accompagner dans leurs premiers développements dans le e-commerce. Cette suite contient les applications phares d'IBM sur la plate-forme Linux : le logiciel de messagerie Lotus/Domino, le serveur d'applications WebSphere, ou encore la base de donnée DB2. Choix réaliste dans le cadre des PME, les clients de ces services sont donnés dans des versions pour MS-Windows. Le tout est vendu au prix canon d'environ 500 US\$ pour les entreprises de moins de 100 personnes, alors que l'achat séparé de ces différents logiciels reviendrait à 3 600 US\$. ●

<http://www.ibm.com/>

L'AFUL

joue la démonstration par l'absurde

Afin de démontrer la nuisance potentielle d'une extension du système de brevet aux logiciels, l'Association Francophone des Utilisateurs de Linux (AFUL) vient de breveter les procédures de passage aux 35 heures!

La demande officielle a été faite le 13 novembre à l'INPI.

À l'heure actuelle, personne ne peut juger de la valeur légale d'un tel brevet. Mais l'AFUL compte bien le défendre jusqu'au bout! Le moment choisi pour cette provocation ne doit rien au hasard : du 20 au 29 novembre a lieu à Munich une conférence européenne sur la question de la réforme du traité de 1973 qui institua l'actuel bureau européen des brevets. Si l'affaire fait sourire, elle soulève pourtant une question lourde de conséquence pour l'avenir de l'informatique européenne. En s'alignant sur le modèle américain des brevets, L'Union européenne risque d'ouvrir une boîte de Pandore fatale pour sa jeune industrie High-tech. Gageons que les trublions du logiciel libre n'en resteront pas là !

www.aful.org/

La Playstation 2 est-elle un ordinateur ?

Pour bénéficier d'une détaxe, SONY tente un coup de poker encore inédit : faire passer sa console pour un ordinateur !

Sony a implémenté une version maison du langage Basic adapté à sa console de jeu Playstation 2 (PS2). Le Yabasic (pour «Yet Another Basic») est-il

Pour son utilisation, il faudra disposer d'un clavier convenant aux ports USB de la machine. Les programmes ainsi conçus seront sauvegardés sur les cartes mémoires de la PS2.

consoles devraient alors inonder le marché. Si le coup de poker de PS2 réussit, il y a fort à parier que les prochains produits



Duty free ?

Aujourd'hui, la direction des douanes anglaises refusent de se prononcer. De son côté, Sony maintient le lancement européen de sa console pour le 24 novembre. 500 000

de Sega, Nintendo et autres Microsoft devraient rapidement concevoir leur basic spécial console. ●
Le site de la PS2 : <http://www.playstation-europe.com/home/home1.jhtml>



destiné aux programmeurs de jeux amateurs ? On est en droit d'en douter, et l'affaire semble se résumer à une histoire de gros sous. Actuellement en pourparlers avec les autorités britanniques, Sony espère échapper à la taxe de 2,2% qui frappe les appareils de jeux vidéo. En obtenant la dénomination d'ordinateur à part entière, la PS2 version européenne augmenterait d'autant sa marge bénéficiaire. Rien, en effet, ne semble indiquer que le constructeur japonais veuille reporter cette détaxe sur le prix public de son appareil, prévu à 2990 Francs.

Pour l'heure, il n'est pas certain que les Playstation 2 vendues en France seront accompagnées du Yabasic. Si tel est le cas, le langage sera fourni sur le CD-ROM de démonstration avec d'autres utilitaires.

E-déogramme

La « Babel numérique » s'élargit encore

L'expansion d'Internet en Asie souffrait encore d'une barrière d'ordre linguistique. L'enregistrement des noms de domaine dans les langues à idéogrammes n'était, jusque-là pas possible. De fait, les adresses web des sites chinois, japonais ou coréens se déclinaient en caractères romains et en chiffres. La société Registrars.com vient d'ouvrir début novembre un service d'enregistrement de noms de domaine ouvert aux idéogrammes. Le succès semble énorme puisque le journal officiel chinois Xinhua parle de 450 000 demandes enregistrées en moins de deux jours ! Chiffre d'autant plus impressionnant que le dépôt de noms de domaine est un phénomène très récent dans ce pays. Du coup, on peut s'interroger sur des problèmes plus proches de nous. Si les idéogrammes sont disponibles, on peut espérer voir apparaître les caractères latins accentués pour les langues latines telle que le français. Le fait est qu'un service de l'Internet Society travaille actuellement sur l'internationalisation des noms de domaine : l'IETF (*Internet engineering task force*). Mais les entreprises spécialisées dans l'enregistrement de noms de domaine semblent tentées de mettre à l'essai leurs propres solutions à ce problème, ce que leur déconseille officiellement l'Internet Society afin de préserver le bon fonctionnement du système actuel. Le DNS doit pouvoir continuer de permettre à tout système, où qu'il se trouve, de résoudre tous les noms de domaine.

<http://www.i-d-n.net/>

Modem

En avant pour le V92

Début 2001, la norme V90 (56 Ko) devrait laisser la place à une nouvelle venue baptisée V92 agréée durant l'été par l'UIT (Union internationale des télécommunications). Les premiers produits sont en cours de fabrication et devraient être disponibles en quantité pour le début de l'année. Seul le débit ascendant (vers le fournisseur d'accès) est amélioré et peut atteindre 48 Ko (contre 33,6 pour les V90). Second avantage, bien plus significatif: cette norme permettra de recevoir

des appels téléphoniques durant une connexion web (comme une connexion ADSL, mais sans simultanéité). Avec un modem V92, l'internaute sera informé d'un appel téléphonique entrant. Il pourra alors interrompre la navigation ou le téléchargement, répondre à l'appel, et poursuivre ses opérations sur Internet, là où il en était auparavant. Enfin, les délais de connexion au réseau sont améliorés (15 s. contre une trentaine avec le V90). Si aucun problème ne se pose



pour les acquéreurs d'un nouvel appareil, les choses sont moins claires pour ceux qui décident d'une mise à jour. Les possesseurs de *soft modems* sont avantagés: une simple mise à jour du logiciel permettra de bénéficier de ces fonctions. US Robotics est le premier constructeur à annoncer la mise à jour de trois modèles 56 Ko: FaxModem, Message Modem et Professional Message Modem.

Chez d'autres, on étudie des possibilités de reprise lors de l'achat d'un modèle V92. ●

Sécurité

Une offre complète pour les PME



L'éditeur de logiciels de cryptographie et sécurité Cryptogram et le spécialiste des solutions d'authentification Safe Data System annoncent la signature d'un accord visant à proposer aux grands comptes et aux PME une solution globale de sécurisation de leurs systèmes d'information. Cette offre intègre deux produits de Cryptogram, *Folder* et *Secure Login* et un de Safe Data System, *Data Remote Access*.

Folder offre de chiffrer et déchiffrer à la volée des fichiers (y compris les messages électroniques) de manière transparente pour l'utili-

sateur tandis que *Secure Login* est une solution d'authentification matérielle permettant d'assurer la sécurité d'accès aux ordinateurs. Ce produit fonctionne en version clé USB, carte à puce ou reconnaissance biométrique. Parallèlement le serveur d'authentification permet de gérer, contrôler et vérifier l'identité des utilisateurs distants souhaitant se connecter au système d'information. La solution complète est proposée au prix d'environ 2 000 francs par utilisateur pour une configuration de 100 postes. Des informations complémentaires peuvent être obtenues sur les sites suivants :

<http://www.cryptogram-fr.com> et

<http://www.safedata.com>. ●

Commerce électronique

Nouvelle version d'ActaWorks

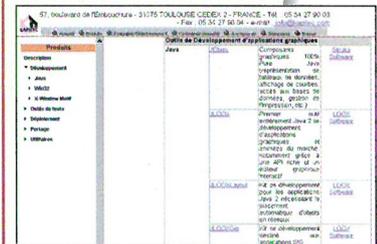
L'éditeur Acta technology, spécialisé dans le développement de bases de données cache pour le commerce électronique présente la version 4.3 d'ActaWorks, son logiciel phare. Cette nouvelle version qui se caractérise par des améliorations de performances et un support multi-utilisateurs s'accompagne d'interfaces avec de nombreux systèmes parmi lesquelles IBM MQSeries et Acta Detail. Ces deux produits permettent notamment de bâtir des solutions de commerce électronique tirant parti de données issues de moyens et grands systèmes IBM. Informations complémentaires à l'adresse www.acta.com. ●

Sur le Fil

Nouvelle start-up à Toulouse

Éditeur java

Vous cherchez les logiciels de Sitraka Software (composants J Class et Jprobe), d'Intuitive Systems (Java Optimizelt), de ZeroG (InstallAnywhere), Loox software, WebPerformance, Object Line : l'éditeur toulousain Kapitec software est la bonne adresse. Cette jeune société créée en avril 2000 par Sandrine



Boquéto-Verdun se spécialise dans la distribution de logiciels Java, d'interfaces homme-machine, dans les tests et le déploiement de ces solutions. Plus d'informations sur le site de Kapitec à l'adresse :

<http://www.kapitec.com>

Erratum

Une erreur s'est glissée dans le numéro 24 de *Programmez!*. Les coordonnées de la société Mutek qui édite le logiciel Bug Trapper sont : Mutek Solutions France 24, rue Jacques Ibert 92300 Levallois-Perret. Avec nos plus humbles excuses.

Développement web

Version 4 de Dreamweaver et FireWorks

Macromedia présente de nouvelles versions 4 de ses environnements de développement Dreamweaver et FireWorks. Ces produits se caractérisent par une flopée de nouvelles fonctions, mais surtout par une meilleure intégration et des capacités d'extension étendues. L'environnement Dreamweaver s'accompagne d'UltraDev 4 destiné à la création d'applicatifs serveurs. Ce logiciel est actuellement le seul capable de générer de l'ASP Microsoft, du JSP Sun et du CFML (Allaire coldFusion). Les versions françaises, distribuées par Apacabar et Feeder, seront disponibles au début de l'année 2001. Le prix est d'environ 5 500 francs pour l'ensemble. <http://www.macromedia.com>.

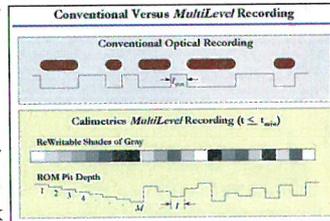
Matériel

Des CD à plusieurs couches

TDK et Calimetrics vont lancer dans le courant de l'année 2001 un nouveau format de disque enregistrable trois fois plus rapide et proposant une capacité de stockage trois fois plus importante que le format CD-R/RW. Les nouveaux produits proposés seront des lecteurs enregistreurs construits selon le standard CD-R/RW. Ils reposent sur l'utilisation de la technologie MultiLevel Recording (ML) Recording de Calimetrics. La capacité sera de 2 Go à une vitesse de 36X sur un disque vierge spécialement formaté ML ou 700 Mo sur

un disque CD-R/RW standard. Selon les deux sociétés, le prix de commercialisation de ces appareils devrait être très légèrement supérieur à celui des lecteurs-enregistreurs classiques. Si TDK et Calimetrics affirment que le format ML n'est pas un remplacement du DVD, il reste possible que cette technologie nuise à son développement.

www.tdk-europe.com
www.calimetrics.com



Utilitaires

Découper un disque en toute sécurité

Powerquest présente une nouvelle version de PartitionMagic, célèbre logiciel de partitionnement de disques durs. Cette nouvelle version ajoute le support natif des systèmes d'exploitation Windows 2000 et Windows ME. La possibilité de partager des partitions FAT et FAT32 et la récupération de partitions FAT, FAT32, Linux ext2 et NTFS préalablement supprimées figurent également

parmi les nouveautés. La version professionnelle permet aussi de copier ou déplacer des partitions depuis ou vers une machine distante par l'intermédiaire du réseau TCP/IP. Cette fonction sera grandement appréciée des responsables informatiques qui pourront dès lors préparer un type particulier de formatage des disques lequel sera ensuite déployé simplement sur l'ensemble du réseau. Notons



PROVEN SOLUTIONS FOR STORAGE MANAGEMENT

également la possibilité de déplacer des fichiers entre deux partitions ou encore de transformer le système de partitionnement sans perte de données. Cette nouvelle version, disponible en français, est proposée au prix public de 590 francs TTC. La mise à jour d'une version antérieure coûte 390 francs H.T. Plus d'informations sur le site de l'éditeur: <http://www.powerquest.com>.

■ La version 6 du serveur web One to One Enterprise de Broadvision supporte J2EE et s'ouvre également aux serveurs d'applications de BEA, IBM, Iplanet et Iona.

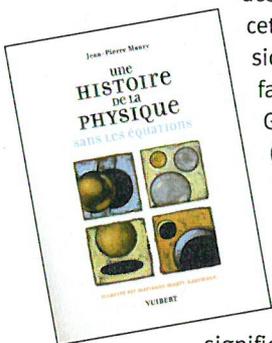
■ Mercury interactive propose Topaz3.0 intégrant un nouveau composant de surveillance. Ce composant permet d'identifier les problèmes de performances des FAI et des serveurs web. Notons que la société propose également 7 jours de surveillance gratuite de votre site internet.

■ Hewlett Packard et RedMessage s'associent pour l'intégration de solutions de communication mobile dans certains produits HP. L'objectif est également de développer de nouvelles solutions basées sur l'Unix HP.

■ Alladin présente la version bêta de eToken Enterprise, une solution d'authentification à deux facteurs basée sur une signature des courriers électroniques et des fonctions de chiffrement. D'après l'éditeur, cette solution est « plug & play » et s'intègre facilement dans les systèmes de messagerie les plus courants.

Physique facile

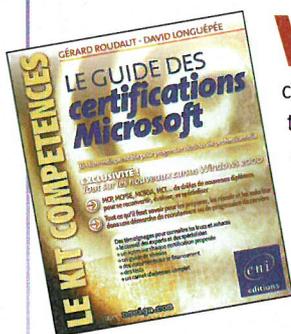
La science en général et la physique en particulier vous ont toujours intéressé mais vous vous souvenez avec douleur des résultats désastreux que vous obteniez dans cette matière. Vous aimez la physique mais la physique ne vous aime pas. Ce livre est véritablement fait pour vous. Newton, Volta, Faraday, Maxwell, Galilée, Aristote... Ils sont tous là, jusqu'en 1932 (un grand millésime selon l'auteur) avec le neutron, le positon, le deutérium et le cyclotron. Vous découvrirez leur histoire, l'histoire de leurs inventions et tout ceci – comme le titre le précise – sans avoir à subir une seule équation, ni même froncer les sourcils pour comprendre la signification des phrases. Vous découvrirez les anecdotes et les concours de circonstances qui ont parfois amené les plus grandes découvertes. Vous apprendrez le pourquoi de la pomme de Newton et tant d'autres histoires. C'est amusant, c'est instructif et c'est à conseiller aux adolescents comme aux adultes. Pour notre part, ce livre est déjà parti sur notre table de chevet, et assurément pas pour vaincre les insomnies. ●



Une histoire de la physique sans les équations, Jean Pierre Maury. Editions Vuibert. 150 francs.

Une nouvelle corde à son arc

Vous êtes Jeune diplômé d'une école d'informatique ou encore vous cherchez une reconversion profitable, la certification Microsoft est un plus appréciable sur un curriculum vitae et un moyen de décrocher rapidement un emploi. Comment s'y retrouver dans le maquis des acronymes? quel est le contenu de telle ou telle formation? Comment s'y préparer et quels sont



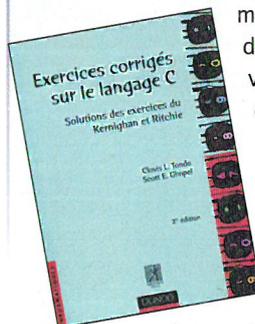
les pré-requis? ce guide vous permettra de choisir le meilleur cursus compte tenu de vos objectifs et vous servira de point de départ dans la recherche des informations nécessaires à passer les examens. Si cet ouvrage constitue un bon point de départ, on regrettera que les thèmes soient souvent survolés et que certains d'entre-eux soient présentés uniquement en anglais, l'effort de traduction n'ayant même pas été fourni. Bref, l'idée est intéressante mais le résultat pas vraiment concluant. ●

Le guide des certifications Microsoft, Gérard Roudaut - David Longuépée. Edition ENI. 150 francs.

Progresser en C

Exercices corrigés sur le langage C, Clovis Tondo et Scott Gimpel. Éditions Dunod. Environ 150 francs.

Conçu en grande partie par Brian Kernighan et Denis Ritchie au début des années 80, le langage C reste encore aujourd'hui le langage de référence pour bon nombre de programmeurs. Un nombre tout aussi important d'apprentis ont découvert ce langage à travers le livre publié par les deux auteurs et connu sous l'abréviation K&R. Malheureusement, les exercices proposés ne contenaient pas de solution. Clovis Tondo et Scott Gimpel reprenent ce principe mais offrent, quant à eux, les solutions aux exercices, en conservant la progression dans l'apprentissage. Bref, ce livre est

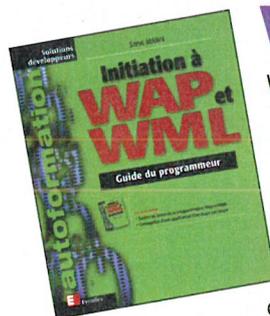


à conseiller à tous les programmeurs en herbe ainsi qu'à ceux susceptibles de passer des tests d'embauche, pas mal de sociétés s'inspirant de ces exercices pour tester les connaissances de leurs postulants. ●

Débuter en WAP

Initiation à WAP et WML, Steve Mann. Éditions Eyrolles. Prix : environ 180 francs.

Vous souhaitez développer sur les mobiles et découvrir les spécifications WAP et WML. Cet ouvrage présente les bases du langage WML et de son pendant pour le script WML Script. En progressant dans le livre, vous apprendrez à construire une véritable application en s'inspirant de WorldFAQ développée par phone.com. Didactique et clair, cet ouvrage est fort pratique mais nous regrettons l'absence de CD qui oblige à ressaisir l'ensemble des programmes présentés dans l'ouvrage. Sans faire de vous un spécialiste de ce langage, cette initiation vous donnera les clés d'un langage potentiellement utilisé par 1 milliard d'utilisateurs de téléphones sans fil à l'horizon 2003. ●



regrettons l'absence de CD qui oblige à ressaisir l'ensemble des programmes présentés dans l'ouvrage. Sans faire de vous un spécialiste de ce langage, cette initiation vous donnera les clés d'un langage potentiellement utilisé par 1 milliard d'utilisateurs de téléphones sans fil à l'horizon 2003. ●

JBuilder 4 la version

café très fort

Avec la nouvelle version de son environnement de développement Java, Inprise propose un outil proche de la perfection.

Frédéric Mazué
frederic.mazue@wanadoo.fr



Dans cet article nous testons la version «Entreprise» du produit. Le lecteur voudra bien garder à l'esprit que toutes les fonctionnalités dont nous parlons ne sont pas forcément incluses dans les versions «Standard» et «Professionnelle». Les fonctionnalités de chaque version sont détaillées sur le site d'Inprise à <http://www.inprise.fr>.

La configuration minimale requise pour la version Entreprise est :

Matériel :

- 128 Mo de Ram.
- 250 Mo d'espace disque au minimum.
- Un lecteur de Cd-Rom.
- Une résolution SVGA ou supérieure (800*600 256 couleurs).
- Une souris (ou analogue).

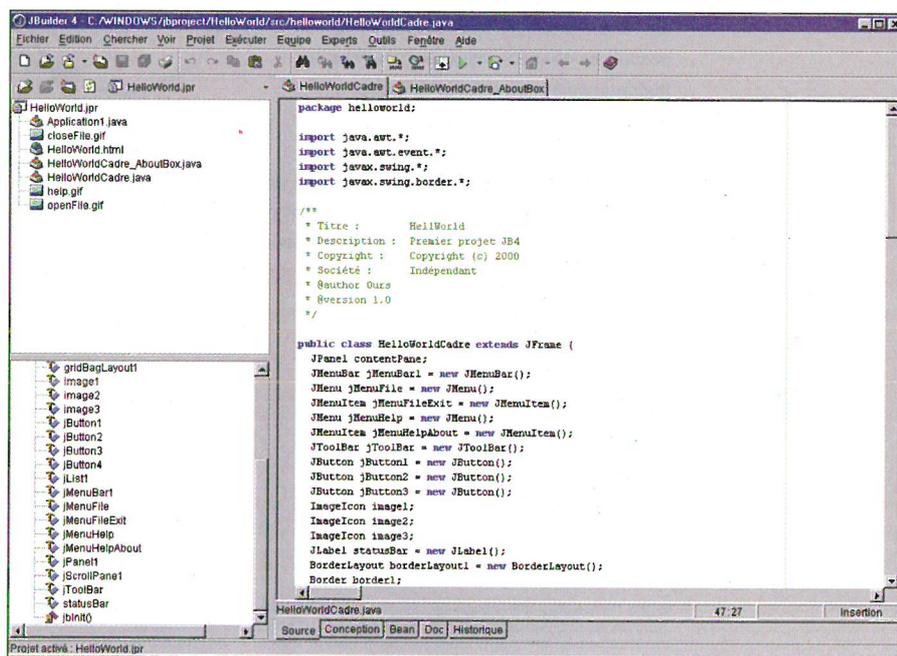
- Intel Pentium II/233 MHz ou compatible pour machine PC.
- Microsoft Windows 98, 2000 ou NT (SP3)
- Linux Red Hat 6.2 ou plus, Mandrake 6.x ou plus, Caldera Open Linux 2.3, SuSE 6.3 ou plus
- Solaris ULTRASparc 2, Solaris 2.6 ou 7 (2.7)

On le voit, JBuilder 4 se présente comme un outil multi-plates-formes. JBuilder 4

est écrit entièrement en Java. Nous avons testé le produit sur un ordinateur PC avec Microprocesseur Intel Pentium II/233Mhz avec 160 Mo de Ram. La machine de test est une machine *double-boot* : Windows 95 d'une part, en dépit de la recommandation d'Inprise, et Linux Mandrake 7.0 d'autre part. Nous n'avons pas testé le produit sur plate-forme Solaris. Mais, l'ayant fait sous Linux c'est-à-dire un système UNIX, tout comme Solaris, et considérant que Sun Microsystems ne peut pas être mieux placé pour fournir une JDK performante sous Solaris, nous pensons qu'il n'y a aucune raison de craindre que JBuilder 4 ne donne pas satisfaction sur cette plate-forme.

Ouvrons la boîte

Nous sommes convaincus que tout outil de développement sérieux doit comporter une documentation papier convenable. Fidèle à son habitude Inprise fournit une documentation de qualité composée de 4 manuels. Un manuel de présentation du langage Java et de prise en main du produit, un manuel dédié à l'élaboration des applications de bases de données, un manuel consacré à JDataStore, un outil de base de données écrit entièrement en Java et enfin un manuel dédié à l'écriture d'applications pour l'entreprise, c'est-à-dire des applications distribuées mettant en œuvre RMI, CORBA, EJB, etc. Cette documentation est très claire et très soignée. Il est manifeste qu'elle est le fruit d'un réel effort de pédagogie visant à faciliter au maximum l'apprentissage de JBuilder 4.



► Ecran 1 : Vue d'ensemble de l'environnement.

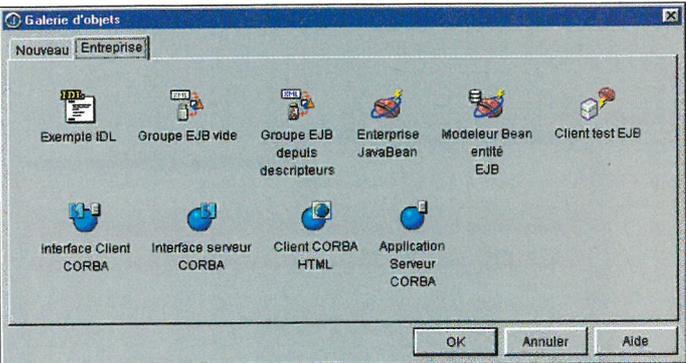
Chaque fonctionnalité de Jbuilder 4 est couverte par un didacticiel très détaillé. Nous avons eu également la bonne surprise de trouver un Delphi 4 et un C++Builder 4 dans le «paquetage».

Installation du produit

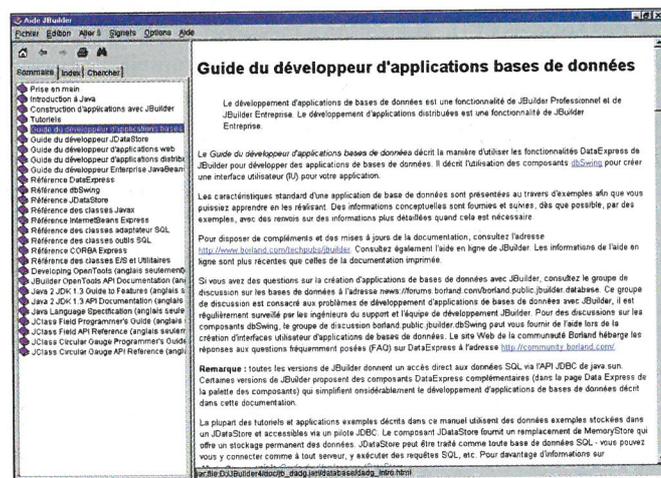
La procédure d'installation et la première utilisation des outils sont assez pittoresques et vous donneront l'impression de participer à une émission interactive de Fort-Boyard. Sous Windows l'installation se passe bien, mais celle-ci achevée, vous devez partir en quête de clés d'enregistrement et autres clés de licence sagement dissimulées dans les endroits les plus hétéroclites. La pochette de CD pour JBuilder lui-même, le site d'Inprise pour Inprise Application Server, la documentation électronique pour JData-

produit sans être propriétaire du serveur X, faute de quoi vous pourriez vous heurter à des problèmes vraiment désagréables. Pour être propriétaire de X, démarrez votre Linux en mode console c'est-à-

dire le niveau 3 pour Red Hat et Mandrake ou le niveau 2 pour une SuSE. Connectez-vous ensuite sous le nom d'utilisateur pour lequel vous souhaitez utiliser le produit, puis lancez le serveur X par la commande **startx**. Nous vous conseillons très vivement de procéder de la même manière MÊME si vous voulez installer JBuilder 4 en tant que **root**.



► Ecran 3 : JBuilder 4 est riche en wizards, comme il se doit.



► Ecran 2 : La documentation électronique de JBuilder 4.

Store et la documentation papier pour Interbase. Tout cela est assez fastidieux. Nous comprenons fort bien le souci légitime que peut avoir Inprise de vouloir protéger ses produits. Il n'empêche que si les fameuses clés avaient été regroupées dans un même endroit cela n'aurait pas été plus mal.

L'installation sous Linux peut se révéler plus difficile. Inprise a choisi d'utiliser l'outil d'installation multi-plates-formes «InstallAnywhere». Ce choix nous paraît malheureux. «InstallAnywhere» ne semble pas avoir beaucoup d'atomes crochus avec les pingouins. Voici ce qu'il faut faire pour être à l'abri des ennuis. Tout d'abord, il ne faut pas installer le

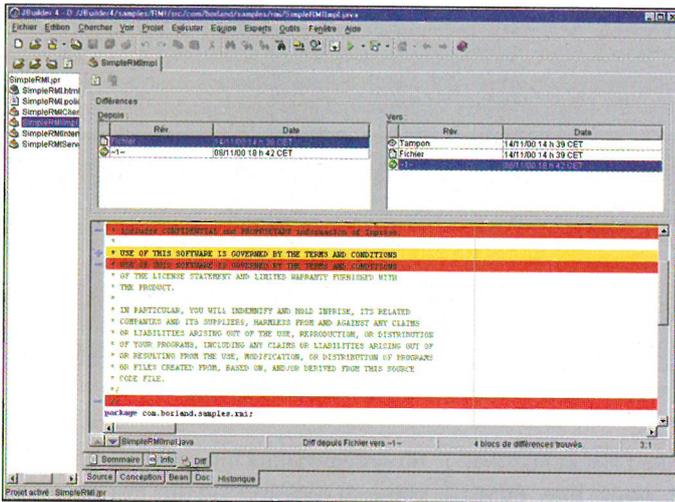
du système Linux lui-même sur votre machine. Si vous avez fait une installation de type serveur, il y a de grandes chances pour que votre répertoire **/tmp** soit situé sur une partition largement dimensionnée. Pas de difficulté dans ce cas. Si vous avez fait une installation de type développement et suivi les recommandations de Red Hat, votre partition racine est assez petite et le répertoire **/tmp** s'y trouve. Dans ce cas de figure et faute de place, l'installation peut échouer sans rien dire ou en émettant un message d'erreur n'ayant rien à voir avec le problème. Si cela vous arrive, il vous suffit de créer un lien symbolique pour faire pointer le répertoire **/tmp** sur un endroit plus spacieux au moins le temps de l'installation et le problème est réglé. Notez bien que nous ne parlons pas ici des 250 Mo d'espace disque occupés par JBuilder après installation, mais bien de la capacité du répertoire **/tmp** requise pendant l'installation.

Il décompresse ensuite le fichier en question puis désarchive ce qui se révèle être la JDK 1.3, ceci toujours dans le répertoire **/tmp**. Inprise déclare que 50 Mo disponibles sur le répertoire **/tmp** suffisent à assurer le bon déroulement de l'opération d'installation. L'installateur teste d'ailleurs l'existence de cet espace de 50 Mo au début de la procédure et vous avertit en cas de besoin. Le problème est que l'installation demande en réalité beaucoup plus d'espace dans votre répertoire **/tmp**. Au moins 150-200 Mo, nous ne savons pas exactement. Toujours est-il que ceci peut provoquer l'échec de l'installation de JBuilder 4 selon la procédure employée pour celle

de votre répertoire **/tmp**. Il décompresse ensuite le fichier en question puis désarchive ce qui se révèle être la JDK 1.3, ceci toujours dans le répertoire **/tmp**. Inprise déclare que 50 Mo disponibles sur le répertoire **/tmp** suffisent à assurer le bon déroulement de l'opération d'installation. L'installateur teste d'ailleurs l'existence de cet espace de 50 Mo au début de la procédure et vous avertit en cas de besoin. Le problème est que l'installation demande en réalité beaucoup plus d'espace dans votre répertoire **/tmp**. Au moins 150-200 Mo, nous ne savons pas exactement. Toujours est-il que ceci peut provoquer l'échec de l'installation de JBuilder 4 selon la procédure employée pour celle

L'éditeur et l'environnement

Nous devons le confesser : nous avons été très surpris. Testant le produit sous Windows 95 avec une configuration proche du minimum nous nous attendions à une utilisation très laborieuse du produit. Pas du tout, c'est même exactement le contraire. Le temps de réponse aux actions de l'utilisateur est remarquablement rapide pour une application Java. Cela tient sans doute à plusieurs choses. D'abord l'EDI semble très bien écrit. Ensuite la JDK 1.3 pour Windows de Sun est très performante. De gros progrès ont été faits depuis JDK 1.2.2. Par exemple, les pauses de gc sont peu nom-



► Ecran 4 : Grâce à CVS vous pouvez faire le point sur votre travail d'un coup d'oeil.

breuses et très douces ce qui assure une grande fluidité à l'utilisation. Enfin, le compilateur JIT HotSpot de Sun se débrouille très bien et surtout l'ensemble gère très efficacement la mémoire. Ainsi nous avons toujours travaillé très à l'aise avec nos 160 Mo de Ram même lors des essais concernant les applications distribuées. Le mécanisme de mémoire virtuelle de Windows est vraiment très rarement sollicité ce qui n'était pas le cas avec les versions précédentes de JBuilder. Sous Linux, on emploie une JDK 1.3 d'IBM avec compilateur JIT. Les performances globales sont un petit peu moins bonnes, parce que, semble-t-il, les performances de la JDK 1.3 d'IBM le sont également, surtout au niveau de la gestion de la mémoire. Toutefois, les performances sont tout-à-fait honorables.

Tout a été pensé pour que le développeur travaille, seul ou en équipe, avec le maximum de confort dans un environnement remarquablement stable (voir l'écran 1).

JBuilder 4 vient avec une documentation électronique très soignée. Nous regrettons, malgré ces éloges, que quelques parties propres à JBuilder comme l'API Open Tools n'aient pas été traduites en français (voir l'écran 2).

Grâce aux performances des JDK 1.3 sous-jacentes la documentation peut être consultée rapidement et se révèle une aide véritable pour le programmeur.

Il est possible d'accéder à tous les outils du boîtier depuis l'éditeur. Il est même possible de personnaliser à outrance celui-ci grâce à l'API Open Tools. Entière-

ment écrite en Java, l'API Open Tools consiste en un ensemble de classes et d'interfaces servant de bases à l'écriture d'applications venant se greffer dans l'EDI. Vous pouvez ainsi ajouter des menus, un lecteur d'e-mail, etc.

L'éditeur dispose bien sûr de tous les wizards possibles et imaginables, (voir l'écran 3), mais il y a en plus de véritables trouvailles dans cette version. Par exemple le gestionnaire de bibliothèques est vraiment intéressant. Les bibliothèques sont des collections de chemins vers des classes, des sources et des fichiers documentation. JBuilder 4 y recherche tout élément nécessaire à l'exécution d'un projet, à la navigation dans les sources ou à la conception visuelle.

L'environnement supporte également CVS (Concurrent Version System). Celui-ci est un outil remarquable pour le travail en équipe. Même pour un développeur isolé CVS trouve son utilité. Un simple clic sur l'onglet historique de la fenêtre d'édition et vous pouvez faire le point sur votre travail d'un coup d'oeil (voir l'écran 4).

Dans JBuilder 4 un constructeur d'archives vient remplacer très avantageusement l'ancien expert de déploiement (voir l'écran 5). Le constructeur d'archives est très pratique, très souple et il a la bonne idée d'être capable de créer des manifestes.

Une fois l'archive définie, un nœud apparaît dans l'arborescence du projet. Vous pouvez ainsi modifier facilement le contenu de l'archive puis la reconstruire

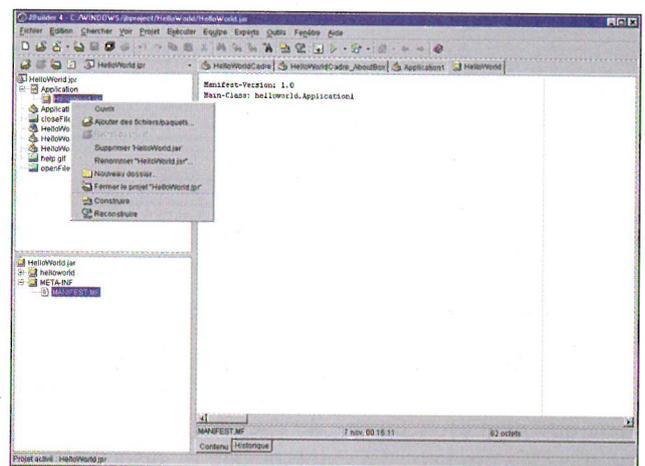
directement à l'intérieur du projet ce qui est une fonctionnalité vraiment pratique (voir l'écran 6).

Signalons enfin que nous avons particulièrement apprécié la fonctionnalité d'achèvement de code qui est très rapide et très bien faite. Puisqu'il faut aussi critiquer un petit peu, nous regretterons qu'à l'intérieur de l'outil de conception visuelle la prise en charge du gestionnaire de mise en forme « GridBagLayout » soit toujours aussi peu intuitive. Il faut vraiment s'y habituer avant d'en tirer profit.

Le cycle écriture compilation débogage

Nous avons déjà un éditeur performant et productif. À cela s'ajoute un compilateur qui s'est avéré rapide et fiable lors de nos essais. Le compilateur émet en français des messages d'erreur explicites le cas échéant. Le gestionnaire des projets assure une construction rapide des applications.

Le débogueur est complet. Il permet de visualiser les threads, les piles d'appels,

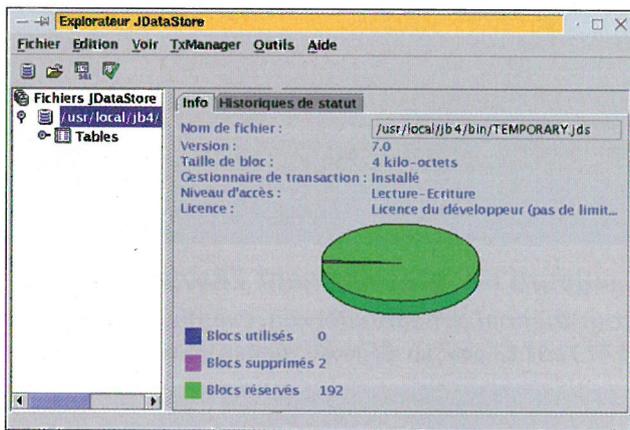


► Ecran 5 : Une fois l'archive définie, vous pouvez modifier son contenu directement depuis le gestionnaire de projets.

le suivi de données, les classes chargées, les points d'arrêts, conditionnels ou non, etc. Nous l'avons également trouvé parfaitement fiable. Il est possible de faire du débogage distant multi-JDK et surtout multi-plates-formes. Quelle opportunité que de pouvoir travailler avec un seul et même outil sur plusieurs plates-formes!

Une note à l'intention des utilisateurs Linux qui n'ont pas de réseau : le débogueur de JBuilder 4, Java oblige, utilise abondamment le protocole TCP/IP. Il faut donc avoir une interface réseau minimum en état de marche sur le système, même si votre machine n'est connectée à aucune autre. Une simple interface *loop-back* fera l'affaire. Il suffit que vous ayez une ligne telle que :

```
ifconfig lo 127.0.0.1
```



> Ecran 6 : L'explorateur JdataStore vous permet de manipuler manuellement vos bases de données SQL (copie d'écran Linux).

dans votre séquence de démarrage pour pouvoir utiliser localement le débogueur sous Linux.

JDataStore

JDataStore est un outil de bases de données au standard SQL 92 entièrement écrit en Java et peu encombrant. (voir l'écran 7).

JDataStore peut gérer des flux « table » et des flux fichiers. Les flux « table » qui peuvent être des tables de bases de données complètes créées avec JDBC ou DataExpress ont également la faculté de comprendre des données de table en mémoire cache en provenance d'une source externe telle qu'un serveur de bases de

données.

Les flux de fichiers quant à eux, supportent les objets Java sérialisés. Nous sommes ici en présence d'un très bon outil.

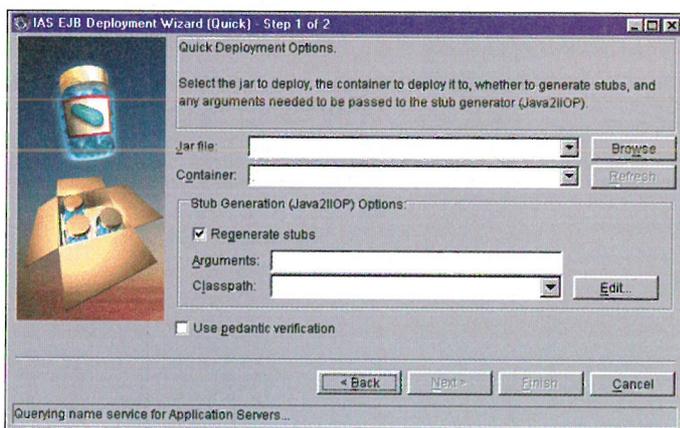
Applications d'entreprise et distribuées

JBuilder 4 dispose d'une panoplie réellement impressionnante de fonctionnalités permettant l'écriture rapide d'applications distribuées en tous genres. Nous apprécions tout particulièrement la prise en charge de CORBA. Il est à peine exagéré de dire que l'écriture d'une application client-serveur CORBA sous JBuilder 4 se résume à l'écriture des interfaces IDL. JBuilder 4 vient avec l'ORB

CORBA Visibroker d'Inprise, mais vous avez la possibilité d'en utiliser un autre, OrbixWeb, par exemple.

Entreprise JavaBeans

Nous n'avons pas la place de décrire tout ce qu'offre JBuilder 4 dans le monde des JSP, Servlets, RMI, XML et consorts. Signalons malgré tout Tomcat qui est l'implémentation des spécifications Java Servlet 2.2 et JavaServerPages



> Ecran 7 : L'expert de déploiement des EJB.

1.1 pour utilisation avec le serveur Apache. Faisons une mention spéciale pour la prise en charge des Entreprise Java Beans. Les EJB sont des composants réutilisables destinés à une exécution en environnement distribués. À l'instar de CORBA, les EJB déchargent les programmeurs des soucis de bas niveau. En outre, les EJB présentent l'intérêt d'être portables et réutilisables. Enfin, travailler avec les EJB comporte l'avantage indéniable de permettre de diviser le développement d'applications distribuées en tâches spécifiques.

Dans ce contexte, JBuilder 4 améliore encore la productivité du programmeur grâce à ses nombreux *wizards* et son expert de déploiement.

Le modèleur de Bean permet de mettre un composant en relation avec une base de données en quelques clics de souris.

En conclusion

JBuilder 4 Entreprise est un outil d'une extrême qualité. Multi-plates-formes, il est une solution de tout premier ordre pour la création d'applications d'entreprise en langage Java 2, particulièrement en environnement hétérogène. ■



JBuilder4

Éditeur : Inprise/Borland

Les prix :

Foundation : gratuit en téléchargement. 345 F. H.T. chez les revendeurs

Pro : 5 995 F. H.T. 2 495 F. H.T. MAJ

Entreprise : 19 495 F. H.T. 14 495 F. H.T. MAJ



Les Plus : - véritable outil multi-plates-formes
- performances
- documentation



Les Moins : - l'installation sous Linux
- système de clés peu pratique

Appréciation :

Documentation	★★★★★
Installation	★★★★★
Éditeur	★★★★★
Compilateur	★★★★★
Débogueur	★★★★★
Fonctionnalités Entreprise	★★★★★
Confort d'utilisation	★★★★★

Pour gérer les données, XML utilise des fichiers ayant l'extension `.xml`.

Gérer la présentation des données avec XSL

Contrairement à HTML qui faisait un amalgame entre données et présentation, XML permet de séparer les genres.

Pour gérer la présentation, un langage spécifique existe : c'est XSL (*eXtensible Style Language*). Par analogie, XSL est à XML ce que CSS (*Cascading Style Sheet*) est à HTML.

Les fichiers XSL disposent quant à eux d'une extension `.xsl`.

C'est dans les fichiers de présentation que sont gérés les couleurs, les tableaux, les polices de caractères ...



➤ Denis Le Guiff est ingénieur Epita et expert LotusNotes dans une SSII partenaire de Lotus et spécialisée dans le groupware et les nouvelles technologies (Coheris/Atix). Il est Certified Lotus Professionnal en développement et en administration pour les versions 4 et 5 de Notes/Domino et a participé au développement de plusieurs intranets dans de grands groupes internationaux. Il est actuellement chef de projet dans une grande banque française. Il est joignable à l'adresse denisleguiff@hotmail.com.

Nous avons développé un masque et une vue permettant de saisir et de présenter un catalogue produits. Chaque fiche produit se composait alors des champs suivants :

- Nom,
- Marque,
- Modèle,
- Référence,
- Prix,
- Quantité.

Ce sont donc des fiches produit de ce type que nous allons diffuser en XML *via* le Web. Il est important de noter qu'aujourd'hui seul le navigateur Microsoft Internet Explorer dans sa version 5 supporte les différents types de fichiers liés à XML (XML, XSL et DTD) que nous générerons à l'aide d'agents écrits en LotusScript. Les fichiers qui seront utilisés par le serveur pour répondre aux requêtes XML seront placés dans le répertoire `Notes\Data\Domino\HTML` du serveur.

Définir un langage pour votre application grâce à la DTD

Le dernier élément notable de l'architecture XML est le langage de définition permettant de décrire la grammaire des balises créées en XML. On peut ainsi décrire chaque nouvelle balise, définir le type d'éléments qu'elle doit contenir, définir si elle est obligatoire ou facultative et définir les relations entre éléments.

Ainsi, en définissant par exemple, une donnée de type adresse, on peut spécifier que cette adresse se compose d'un numéro de rue, d'un type de voie (rue, avenue, allée...), d'un libellé de rue, d'une indication de localité. Cette indication de localité se composant elle-même d'une indication de pays (facultatif), d'un code postal et d'un nom de ville.

Ce langage de définition DTD (*Document Type Definition*) se matérialise dans un fichier ayant l'extension `.dtd`.

Cependant, il n'est pas obligatoire que la DTD soit dans son propre fichier. En effet, la DTD peut aussi être intégrée dans le document XML (*Embedded DTD*).

L'intérêt des DTD externes est que celles-ci peuvent être échangées avec d'autres applications et ainsi être réutilisées. On peut donc par ce biais définir des grammaires propres à un métier, à un type d'activité, à un type d'application...

Plusieurs dizaines de DTD normalisées sont déjà disponibles pour permettre de faciliter l'échange de données inter-applications (EDI).

Mise en œuvre

La mise en œuvre de XML au sein de Lotus se fera à l'aide de la base de gestion de catalogue produits que nous avons exposée dans l'article publié précédemment.

Génération de la DTD

Avant de pouvoir écrire notre application, il est important de définir sa grammaire.

Dans notre cas, elle est relativement simple car les fiches produit ne contiennent que quelques données.

La DTD que nous allons devoir utiliser devra ressembler à celle décrite dans le **listing 1**, page suivante.

Que signifie cette DTD ?

La première ligne indique la version de XML utilisée.

```
<?xml version='1.0' encoding='utf-8'?>
```

La DTD indique ensuite que notre application va générer un catalogue produits composé de fiches produit (de une à plusieurs fiches)

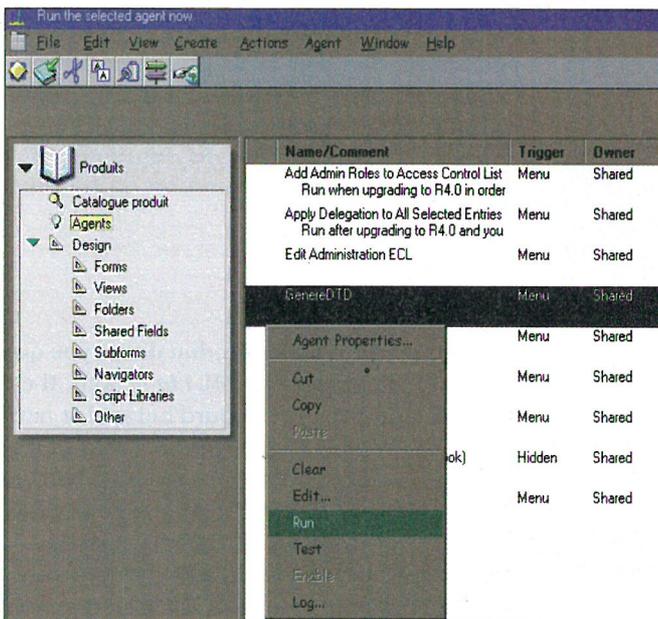
```
<!ELEMENT CatalogueProduit (InfoProduit+)>
```

La troisième ligne stipule que chaque fiche produit se compose de six données qui sont les champs présents dans le masque Notes (Nom, Marque, Modèle, Référence, Prix, Quantité).

```
<!ELEMENT InfoProduit (NomProduit,MarqueProduit,ModeleProduit,ReferenceProduit,PrixProduit,QuantiteProduit)>
```

Les lignes suivantes indiquent que les données de type `NomProduit`, `MarqueProduit`, etc. sont des données de base que l'application fournira.

```
<!ELEMENT NomProduit (#PCDATA)>
<!ELEMENT MarqueProduit (#PCDATA)>
<!ELEMENT ModeleProduit (#PCDATA)>
<!ELEMENT ReferenceProduit (#PCDATA)>
<!ELEMENT PrixProduit (#PCDATA)>
<!ELEMENT QuantiteProduit (#PCDATA)>
```



> Écran 2 : il suffit de cliquer pour activer l'agent.

Il est évident que nous aurions pu nous contenter de générer manuellement ce type de fichier mais il nous a semblé qu'il était judicieux de le faire à l'aide d'un agent LotusScript de façon à disposer dans l'application de toutes les ressources nécessaires à son fonctionnement. Nous allons donc créer un agent dans le panneau des agents de la base **Produits.nsf** (fournie dans le CD-ROM du magazine avec la totalité des listings) comme le montre l'écran 1 page précédente.

L'agent se nommera **GenereDTD**. Il contiendra du code LotusScript uniquement dans la routine **Initialize**.

Le listing 2 présente le contenu de cet agent dans lequel nous allons utiliser un nombre entier (**fileNum**) qui nous servira de *file descriptor* pour accéder à un fichier comme l'indiquent les deux lignes suivantes.

```
Dim fileNum As Integer
fileNum% = Freefile()
```

Nous ouvrons ensuite le fichier **produit.dtd** en accès lecture ce qui a pour effet de créer le fichier si celui-ci n'existe pas déjà.

```
Open «:\notes462\data\domino\html\produit.dtd» For Output Access Write As fileNum%
```

Il ne reste plus qu'à générer le contenu de la DTD en utilisant des commandes **Print file descriptor** « texte à écrire »

```
Print #fileNum%, «<?xml version='1.0' encoding='utf-8'?>»
Print #fileNum%, «<ELEMENT CatalogueProduit (InfoProduit+)>»
Print #fileNum%, «<ELEMENT InfoProduit (NomProduit,MarqueProduit,ModeleProduit,ReferenceProduit,PrixProduit,QuantiteProduit)>»
Print #fileNum%, «<ELEMENT NomProduit (#PCDATA)>»
Print #fileNum%, «<ELEMENT MarqueProduit (#PCDATA)>»
Print #fileNum%, «<ELEMENT ModeleProduit (#PCDATA)>»
Print #fileNum%, «<ELEMENT ReferenceProduit (#PCDATA)>»
Print #fileNum%, «<ELEMENT PrixProduit (#PCDATA)>»
Print #fileNum%, «<ELEMENT QuantiteProduit (#PCDATA)>»
```

comme le montrent les lignes suivantes (voir le listing 1). Lorsque nous avons fini de générer notre DTD, nous fermons le fichier à l'aide la commande **Close**.

```
Close fileNum%
```

Pour activer cet agent, il suffit de se positionner dans la liste des agents, de cliquer avec le bouton droit de la souris puis de sélectionner l'option **Run** (voir l'écran 2). Nous obtenons alors le fichier **produit.dtd** (fourni sur le CD-ROM)

Génération de la feuille de style

La feuille de style est un élément important, car c'est elle qui permet d'indiquer au navigateur comment présenter les données XML qu'il reçoit. Dans notre base LotusNotes, nous disposons d'une vue (vue **Catalogue produits**, le fichier **produits.xml** est fourni sur le CD-ROM) qui présentait la liste des produits disponibles (voir l'écran 3). Nous allons utiliser le même style de présentation que celui obtenu dans un navi-

Produit	Marque	Modèle	Références	Prix du vendeur	Quantité en stock
Ecran 17 pouces	NEC	MultiSync	570*	5470	6
Hot Printer	AWA	Super Base	AW266	365	33
Imprimante	EPSON	Stylus Color	640	990	10
Scanner	AGFA	Snapscan	1212	690	9
Souris	Logitech	3 boutons Wheel mouse	39T-52	125	10

> Écran 3 : la liste des produits disponibles.

gateur grâce à la génération à la volée du code HTML par Domino (voir l'écran 4).

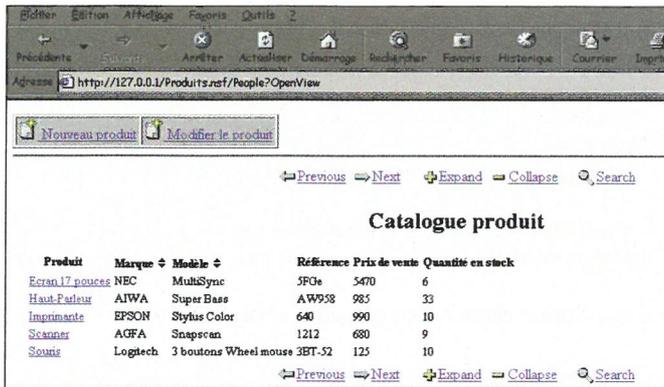
Si l'on visualise le contenu du source de cette page (listing3.txt), on se rend compte que Domino a généré un tableau HTML dans lequel il a mis les données Notes.

C'est ce même principe que nous allons suivre pour générer notre feuille de style (listing4.txt).

Nous constatons dans ce listing que toutes les balises XSL commencent par **<xsl :**. Les autres balises utilisées sont des balises en HTML standard.

Nous déclarons dans un premier temps, la version du langage utilisé et le fait que nous générerons un modèle XSL.

```
<xsl:stylesheet xmlns:xsl='http://www.w3.org/TR/W3-xsl'>
<xsl:template match='/'>
```



> Écran 4 : une présentation « à la Web ».

Nous générons ensuite un tableau HTML composé de 6 colonnes et d'une ligne de titre.

```
<html>
<font face='Verdana' size='3'>
<body>
<table border='0' bgcolor='azure'>
<tr>
<th>Produit</th>
<th>Marque</th>
<th>Modele</th>
<th>Reference</th>
<th>Prix de vente</th>
<th>quantite en stock</th>
</tr>
```

Après, nous utilisons une boucle XSL pour que, pour chaque fiche produit fournie par XML, nous puissions effectuer le même type de traitement, à savoir remplir les cases du tableau à l'aide de chaque type de donnée XML retournée par l'application.

```
<xsl:for-each select='CatalogueProduit/InfoProduit'>
<tr>
<td><xsl:value-of select='NomProduit' /></td>
<td><xsl:value-of select='MarqueProduit' /></td>
<td><xsl:value-of select='ModeleProduit' /></td>
<td><xsl:value-of select='ReferenceProduit' /></td>
<td><xsl:value-of select='PrixProduit' /></td>
<td><xsl:value-of select='QuantiteProduit' /></td>
</tr>
</xsl:for-each>
```

Il ne reste plus qu'à refermer les balises HTML et les balises XSL que nous avons ouvertes.

Adresses utiles De nombreuses informations sur LotusNotes/Domino et sur le langage XML sont disponibles sur le site www.notes.net dans la rubrique IRIS café.

Lors de la dernière grand-messe française du développement LotusNotes qui s'est tenue à Eurodisney, de nombreuses présentations traitaient de XML. On peut télécharger ces présentations sur le site www.lotus.fr rubrique Événements.

```
</table>
</body>
</font>
</html>
</xsl:template>
</xsl:stylesheet>
```

De la même façon que nous avons généré la DTD, nous allons créer un agent (**GenereXSL**) qui va écrire la feuille de style sur le serveur. Les traitements contenus dans cet agent (affectation d'un *file descriptor*, ouverture du fichier en écriture, écriture dans le fichier et fermeture du fichier) sont identiques à ceux effectués précédemment. Le contenu de cet agent est fourni dans le **listing 5 (listing5.txt)**.

Génération du code XML

Nous arrivons maintenant au cœur de l'application, puisque nous allons décrire l'agent générant le code XML.

Nous aurions pu de la même manière que précédemment, générer un fichier ayant l'extension **.xml** dans l'arborescence de fichier LotusNotes. Pour y accéder il aurait suffi de taper le chemin (URL) d'accès au fichier dans un navigateur. Nous avons préféré générer le contenu du fichier « à la volée » afin de minimiser les écritures disque sur le serveur. Les contenus de la feuille de style et de la DTD n'évolueront quasiment pas durant le cycle de vie de l'application. Par contre, les données de l'application (les données XML qui seront publiées) évoluent à chaque mise à jour. Il est donc plus logique de les régénérer dynamiquement de façon à toujours fournir des données correctes.

Pour cela, on crée un nouvel agent (**GenereXML**) et l'on code les commandes LotusScript du **listing 6 (listing6.txt)**. On va dans cet agent déclarer un certain nombre de variables pour accéder à la vue contenant les fiches produits (**vue People**).

```
Sub Initialize
'déclaration des variables
Dim session As New NotesSession
Dim db As NotesDatabase
Dim view As NotesView
Dim doc As notesdocument

'affectation des variables
Set db = session.CurrentDatabase
Set view = db.GetView(«People»)
```

On commence ensuite à générer l'en-tête XML. On utilise pour cela des commandes **Print** qui servent à renvoyer au navigateur qui lancera l'agent les données qui suivent.

```
'définition de l'en-tête XML et indication de la feuille
'de style et de la DTD à utiliser
Print «content-type: text/xml»
Print {<?xml version='1.0' encoding='utf-8'>}
<?xml-stylesheet type='text/xsl' href='>/produit.xsl?>
<!DOCTYPE Produit SYSTEM «>/produit.dtd»>
}
```

Ensuite, on se positionne sur le premier document de la vue, et tant qu'il restera des documents dans la vue, on va balayer celle-ci en générant à la volée les balises XML qui se conforment à la grammaire décrite dans la DTD.

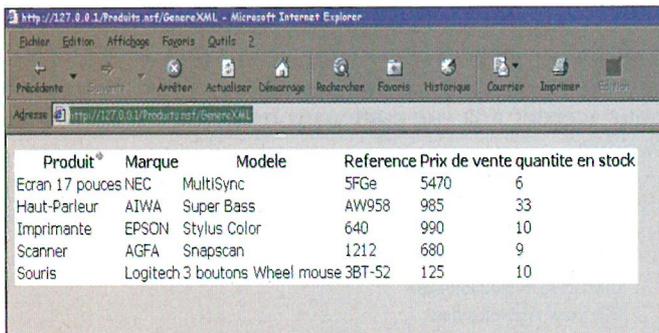
Les balises XML ouvrantes et fermantes encadreront les données Notes (`document.Nom_d'un_champ(0)`).

'récupération de tous les documents de la vue 'afin de générer balises XML

```
Set doc = view.GetFirstDocument
Print «<CatalogueProduit>»
While (Not doc Is Nothing)
  Print {
<InfoProduit>
<NomProduit> + doc.Nom(0) +{</NomProduit>
<MarqueProduit> + doc.marque(0) +{</MarqueProduit>
<ModeleProduit> + doc.modele(0) +{</ModeleProduit>
<ReferenceProduit> + doc.Reference(0) +{</ReferenceProduit>
<PrixProduit> + doc.Prix(0) +{</PrixProduit>
<QuantiteProduit> + doc.Quantite(0) +{</QuantiteProduit>
</InfoProduit>
  Set doc = view.GetNextDocument(doc)
Wend
Print «</CatalogueProduit>»
End Sub
```

Ainsi, pour chaque donnée de notre application nous avons une ligne de générée en XML du type :

```
<Nom_balise>Donnée</Nom_balise>
```



Produit*	Marque	Modele	Reference	Prix de vente	quantite en stock
Ecran 17 pouces	NEC	MultiSync	5FGe	5470	6
Haut-Parleur	AIWA	Super Bass	AW958	985	33
Imprimante	EPSON	Stylus Color	640	990	10
Scanner	AGFA	Snapscan	1212	680	9
Souris	Logitech	3 boutons Wheel mouse	3BT-52	125	10

➤ Écran 5 : l'application terminée, reste à la faire tourner.

Adresses utiles

Durant la dernière Lotusphere d'Orlando plus de 150 présentations ont expliqué la stratégie produit et les implémentations des standards tel que XML au sein des produits Lotus.

Ces présentations sont téléchargeables sur le site www.lotus.com rubrique Events.

Plusieurs cours en ligne sur les fondamentaux de XML sont disponibles.

Citons en particulier, l'université de Syracuse qui dispose d'un cours très bien réalisé et qui est consultable à l'adresse suivante :

www.npac.syr.edu/XML

Pour chaque groupe de données représentant une fiche produit nous encadrons les données par des balises définissant un conteneur :

```
<Nom_conteneur>
  <Nom_balise1>Donnée1</Nom_balise1>
  ...
  <Nom_balise2>Donnée2</Nom_balise2>
</Nom_conteneur>
```

Ce qui donne dans notre cas pour chaque fiche :

```
<InfoProduit>
  <NomProduit>Scanner</NomProduit>
  <MarqueProduit>AGFA</MarqueProduit>
  <ModeleProduit>SnapScan</ModeleProduit>
  <ReferenceProduit>1212</ReferenceProduit>
  <PrixProduit>680</PrixProduit>
  <QuantiteProduit>9</QuantiteProduit>
</InfoProduit>
```

Accéder aux données Notes en XML

L'application est maintenant terminée. Il ne reste plus qu'à l'utiliser.

Pour cela, on lancera la tâche HTTP sur le serveur Notes/Domino de façon à permettre l'accès à un navigateur internet. On saisira l'URL de la l'agent **GenereXML** (du type <http://127.0.0.1/Produits.nsf/GenereXML>).

Le résultat obtenu à l'aide d'Internet Explorer 5 est présenté dans la [capture d'écran 5](#).

Conclusion

De par l'aspect résolument simpliste de cet article, afin d'être didactique, nous avons volontairement écarté de nombreuses spécificités de LotusNotes et de XML. Cependant, cela a permis de démontrer qu'il est viable d'utiliser cette technologie naissante avec ce produit.

Le logiciel LotusNotes évoluant sans cesse, il y a fort à parier que d'ici à quelques mois soit fournit un modèle d'application de génération d'applications XML de façon à ne plus avoir à écrire de code LotusScript pour générer le code XML.

Aujourd'hui, l'état d'intégration de XML dans LotusNotes/Domino, nous a contraint à nous interfacer au plus près des données LotusNotes.

Je pense que dans un avenir plutôt proche, il sera possible de générer la DTD, la feuille de style et le code XML en un simple clic de souris. Les dernières fonctions des produits et technologies Lotus peuvent être étudiées sur le site Internet de Lotus France à l'adresse :

<http://www.lotus.fr> ■

La Bourse


 Sur votre CD-Rom

au cours du Wap...

WAP (Wireless Application Protocol) est un ensemble de protocoles et de langages permettant l'accès à Internet aux appareils mobiles. Comment réaliser une application compatible avec cette technologie? Voyez!

Par Xavier Leclercq
xavier.leclercq@programmez.com

La couche application de WAP est composée du WML (*Wireless Markup Language*) et se base sur XML (*eXtensible Markup Language*), sorte d'équivalent de l'HTML pour les navigateurs, et du WMLScript qui permet de construire une interface utilisateur. Le WML est lu et interprété par un navigateur intégré au portable. De quoi avez-vous besoin pour réaliser une application WAP? En théorie, un simple éditeur de texte suffit, mais bien entendu des kits de développement existent pour réduire le temps de développement. Pratiquement, nous allons examiner comment, sous Linux ou Windows 2000, nous pourrions afficher un cours de Bourse sur un portable WAP...

Connexion du portable via PPP et passerelle Wap

PPP (*Point To Point Protocol*) a été développé pour permettre d'encapsuler et d'envoyer des datagrammes IP via une connexion point à point (RFC-1134). Grâce à cette connexion PPP, et après avoir été autorisé à se connecter par un nom d'utilisateur et un mot de passe, le portable WAP obtient une adresse IP temporaire du fournisseur d'accès. Le micro-navigateur se branchera à un service Internet enregistré par défaut,

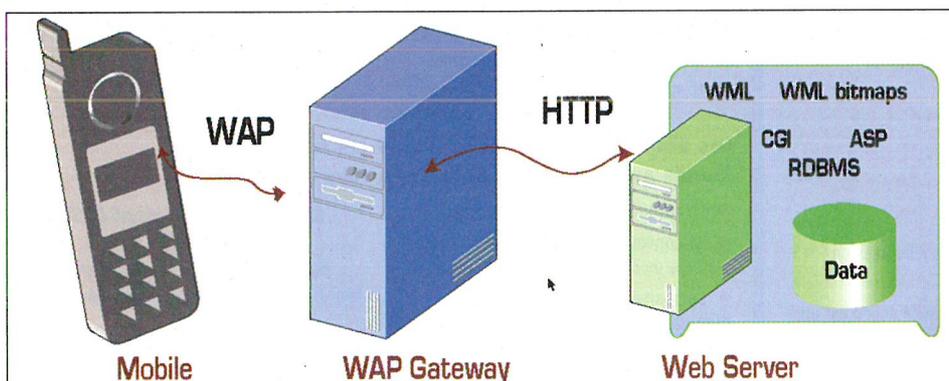
mais vous pouvez utiliser vos propres paramètres pour ainsi accéder à n'importe lequel, en passant par une passerelle WAP voir la figure 1). Celle-ci fonctionne dans les deux sens. D'un côté votre portable s'y connecte, tandis qu'un serveur HTTP (Apache sous Linux ou Internet Information Server sous Windows 2000) le fait de l'autre côté. Le rôle de la passerelle est de relier votre portable à un serveur HTTP. Elle relaye le flux en provenance du serveur HTTP vers le mobile. Dans le sens inverse, des informations, comme un numéro d'utilisateur, par exemple, sont envoyées au serveur HTTP.

Émulez WAP, même les bogues

Les développeurs n'ayant généralement pas un accès direct à une passerelle, vous pouvez vous servir d'un émulateur WAP pour visualiser le résultat à partir d'un poste client de votre intranet. Certains opérateurs en fournissent dans les kits de développement que l'on peut récupérer sur leurs sites. Ces émulateurs sont identiques à de véritables téléphones puisqu'ils poussent le réalisme à émuler les bogues d'un portable! Vous pouvez donc tester facilement vos applications WAP. Mais attention, un site HTML n'est pas forcément visible de la même manière selon le navigateur employé. De même, un site WML sera présenté différemment selon le type de micro-navigateur qu'utilise le portable...

Langage WML

WML utilise des balises «<...>» comme HTML. Les éléments qui se trouvent à l'intérieur des paires de balises peuvent avoir des attributs (le nom de ceux-ci doivent être en minuscules) caractérisés par des valeurs (entre guillemets). Par exemple:
<balise attribut1=>valeur>.



> Figure 1 : Une passerelle Wap permet au portable d'accéder à un service Internet.

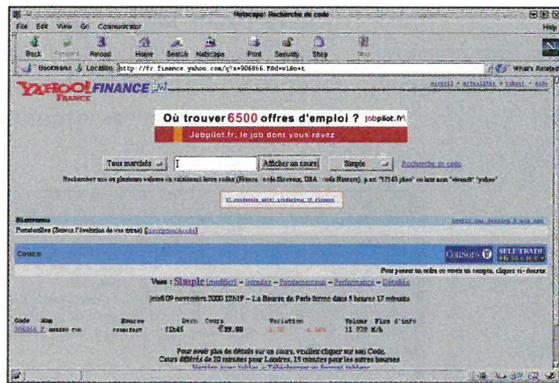
La programmation de variables est aussi possible (sous la forme `$(variable)`).

Une page WML est découpée en paquets de cartes WML (un paquet commence et finit par la balise `<wml>`, et une carte commence et finit par la balise `<card>`).

Le principe est simple. Votre émulateur WAP va accéder à un document WML *via* le serveur HTTP. Il chargera complètement un paquet et vous pourrez naviguer entre les différentes cartes sans devoir solliciter d'autres données. Le micro-navigateur emploie donc une mémoire cache en attendant une nouvelle requête qui provoquera le chargement d'un autre paquet.

Exemple :

```
<wml>
<card id="valeurs en bourse">
...
</card>
<card id="fina">
...
</card>
<card id="cours valeur 2">
...
</card>
<card id="etc.">
...
</card>
</wml>
```



> Écran 1 : Le site Yahoo Finance figure parmi une pléthore de sites d'informations boursières. Aujourd'hui, presque tout le monde peut jouer son Rougon à domicile. Gare aux conséquences !

Informations *Yahoo Finance* (<http://fr.finance.yahoo.com/>). Vous pouvez rechercher (par nom ou par numéro de référence) le cours des actions de votre choix dont l'origine émane des différentes places boursières (environ une vingtaine).

Les informations fournies sont à jour (très détaillées et plus ou moins au quart d'heure près) et des graphiques sont également disponibles. Différentes options d'affichage, comme « Intraday » qui permet de suivre l'évolution d'un cours sur une journée, ou « affichage simple sans tableaux » (que nous utiliserons) vous sont proposées.

En préambule, vous devez connaître les codes des actions que vous désirez suivre. Par exemple pour l'action de « France Télécom » il s'agit pour la Bourse de Paris du code « 13330.PA ». Sur le site de *Yahoo Finance*, le lien vers [Recherche de code](#) vous y aidera. (voir l'écran 1).

Ingrédients

Il nous faut programmer, mais rassurez-vous, un débutant peut y arriver.

Pour élaborer notre « recette boursière », nous avons besoin des ingrédients suivants.

Sous Linux :

- Un Linux fonctionnel en mode console ou une fenêtre Konsole sous X11 ;
- Le *package* « expect » (disponible de base avec la majorité des distributions) ;
- L'archive tar du langage REXX « BREXX » ;
- Une connexion active à Internet et le serveur Apache.

Sous Windows :

- Windows 2000 par exemple ;
- « BREXX », Le langage REXX pour Windows ;
- Une connexion active à Internet et le serveur IIS.

Sachez qu'un paquet WML doit commencer par la déclaration XML suivante :

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM/DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
```

Le plus difficile à mettre en œuvre est donc l'extraction de l'information boursière. La mise en page WML est réalisable automatiquement *via* ASP ou PHP.

Comment extraire une valeur boursière ?

Les indices boursiers sont consultables sur les sites des différentes bourses elles-mêmes, où vous pouvez entre autre suivre le cours de certaines actions. Il y a les grandes places américaines telles que le *New York Stock Exchange* (www.nyse.com), le *Nasdaq* (www.nasdaq.com) et l'*American Stock Exchange* (www.amex.com). En Europe, il y a la Bourse de Paris sur www.bourse-de-paris.fr, la Bourse de Bruxelles sur www.bourse.be, la Bourse de Londres sur www.londonstockex.co.uk, la Bourse de Francfort sur www.exchange.de/fwb/fwb.html. Pour suivre les cours de la Bourse j'ai choisi le site d'infor-

```
genere.sh
#!/bin/bash

proxy=192.168.0.8
port=80

clear

./recup.exp http://fr.finance.yahoo.com/q?s=12007.PA $proxy $port > bourse.txt
./rexx ./bourse.rexx ENTETE
./rexx ./bourse.rexx

./recup.exp http://fr.finance.yahoo.com/q?s=870747.F $proxy $port > bourse.txt
./rexx ./bourse.rexx

./recup.exp http://fr.finance.yahoo.com/q?s=906866.F $proxy $port > bourse.txt
./rexx ./bourse.rexx

./recup.exp http://fr.finance.yahoo.com/q?s=929398.BE $proxy $port > bourse.txt
./rexx ./bourse.rexx
```

Les différentes étapes sont les suivantes :

1. Extraction de la page HTML ;
2. Filtre de la page HTML pour en retirer l'élément qui nous intéresse (le nom de la valeur et son cours) ;
3. Affichage de cette valeur au sein d'une carte d'un paquet WML (je vous laisse réaliser cette dernière partie qui est aussi la plus simple).

Module Visual Basic 6, d'extraction d'une URL sous Windows

Sous Windows, un module Visual Basic peut-être employé (testé sous Windows 2000). Un objet ActiveX est ainsi généré et sera exploitable au sein d'une page ASP.

Attention : vous pouvez programmer ce module dans un autre langage (comme l'autorise d'ailleurs la nouvelle norme de Microsoft ASP+), voir le [listing](#) sur CD-Rom.

Total Expect, m'sieur, sous Linux

Expect est un utilitaire qui a été mis au point par Monsieur Don Libes, auteur d'un livre sur le sujet (*Exploring Expect*, O'Reilly). Expect automatise certaines tâches sous Unix/Linux. Il permet, par exemple, de piloter rapidement Telnet, FTP, rlogin... Nous allons employer Expect pour automatiser l'accès au site *Yahoo Finance via* Telnet.

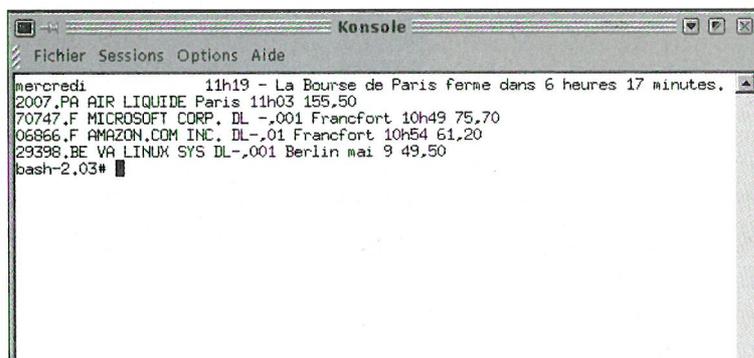
Vous pouvez installer les binaires sous format RPM d'Expect avec YaST sous SuSE. Ou encore vous pouvez compiler les sources ; dans ce cas attention, Expect se base sur Tcl, *Tool*

```
bourse.txt
spawn /usr/bin/telnet 192.168.0.8 80
Trying 192.168.0.8...
Connected to 192.168.0.8.
Escape character is '^]'.
GET http://fr.finance.yahoo.com/q?s=929398.BE HTTP/1.0
```

```
HTTP/1.0 200 OK
Date: Wed, 10 May 2000 09:20:56 GMT
Cache-Control: private
Content-Type: text/html
Connection: close
```

```
<html>
<head><title>Recherche de code</title></head>
<body>
<center>
<table border=0 cellpadding=0 cellspacing=0 width=100%><tr>
<td width=1%><a href=http://fr.finance.yahoo.com><img border=0 src=http://eur.yimg.com/i/fr/g/fin4.gif alt=»Yahoo! Finance« height=40 width=281></a>
```

```
...
- <a href=»/q?s=929398.BE&d=v4»>Performance</a>
- <a href=»/q?s=929398.BE&d=t»>D&eacute;taille&eacute;te,</a>
<p>mercredi 10 mai 2000 11h20 - La Bourse de Paris ferme dans 6 heures 15 minutes.
<table border=1 cellpadding=1 cellspacing=0>
<tr bgcolor=»#dcdcdc»
<th nowrap>Code</th>
<th nowrap>Nom</th>
<th nowrap>Bourse</th>
<th nowrap colspan=2>Dern. Cours</th>
<th nowrap colspan=2>Variation</th>
<th nowrap>Volume</th>
<th nowrap>Plus d'info</th>
</tr>
<tr align=right>
<td nowrap align=left><a href=»/q?s=929398.BE&d=t»>929398.BE</a></td>
<td nowrap align=left><small>VA LINUX SYS DL,.001</small></td>
<td nowrap align=center><small>Berlin</small></td>
<td nowrap align=center>mai 9</td>
<td nowrap><b><img src=http://eur.yimg.com/i/eu/f/e.gif width=11 height=13 border=0 alt=E>49,50</b></td>
...
</center>
</body>
</html>
Connection closed by foreign host.
REXX (Windows 2000 / Linux)
```



Écran 2 : votre portable Wap affiche vos cours de bourse préférés.

2

➤ ➤ **Command Language** qui doit donc avoir été installé!
Le fichier qui commande Expect porte l'extension **.exp**.

```
recup.exp
#!/usr/bin/expect -f

set site [lindex $argv 0]
set proxy [lindex $argv 1]
set port [lindex $argv 2]
set timeout -1
#spawn $env(SHELL)
match_max 100000
spawn /usr/bin/telnet $proxy $port
expect «Escape character is '^」.»
send «GET $site HTTP/1.0\r»
send «\r»
expect «Connection closed by foreign host.»
exit
```

Il est très intéressant de noter que grâce à Telnet, vous pouvez employer un proxy. C'est mon cas, c'est pourquoi j'ai opté pour la solution du pilotage de Telnet *via* Expect.

Script Shell sous Linux

Le script Shell s'invoque en ligne de commande. Si vous avez à votre disposition une ligne ADSL rien ne vous empêche (à l'aide de **cron**) d'exécuter ce script à intervalles réguliers voir le **listing 1**.

C'est au sein de ce script que vous devez renseigner les codes des actions que vous désirez suivre. Les sorties sont sauvées dans le fichier «**bourse.txt**». C'est ce dernier fichier qui sera analysé par notre moulinette «**rexx**».

Pour bien comprendre la suite, voici, sur le **listing 2**, les extraits les plus significatifs du fichier généré.

Rexx existe depuis 30 ans. Son inventeur, Mike Cowlishaw, un génial programmeur d'IBM, l'a implanté en 1970 sur IBM 370.

Il existe pas mal de REXX (commerciaux et libres) sous Linux. J'ai choisi BREXX car c'est celui que j'emploie aussi sous Windows 2000. Dans le même répertoire vous devez avoir l'interpréteur (exécutable du nom de «**rexx**») et son fichier de ressource «**files.r**».

Cette dernière bibliothèque de fonctions spéciales fonctionne de la même manière sous LINUX, Windows 2000, Windows Pocket (CE), NT, OS/2 ou Amiga...

Le programme REXX ouvre le fichier **bourse** et l'analyse ligne par ligne. Dès qu'il tombe sur une marque particulière, il traite les lignes suivantes pour afficher le cours de l'action sur la place boursière (déterminée par l'extension du code, par exemple «**PA**» pour Paris, voir le **listing 3**).

Il ne vous reste plus qu'à ajouter les balises adéquates WML pour que votre portable puisse afficher vos cours de bourse préférés voir **l'écran 2**.

CONCLUSION

Notre «moulinette» aurait pu être écrite en PERL plutôt qu'en BREXX, mais il est plus facile pour un débutant d'aborder le REXX (en outre, la routine **bourse.rexx** est très lisible).

En procédant de la même manière, il n'est guère plus difficile d'afficher d'autres types d'informations comme par exemple la météo. WAP permet d'afficher des informations diverses des plus intéressantes. À vous de l'exploiter au mieux sans modération... ■

3

```
bourse.rexx
/*
  EXTRAIT LE COURS DE L'ACTION DU FICHIER HTML
*/

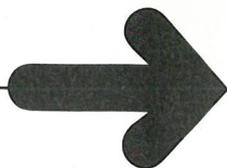
Call load «files.r»

arg ENTREE

nom_du_fichier = «bourse.txt»

/* ouverture en lecture */
HND = OPEN(nom_du_fichier,'r')

I = 0
NBR = LINES(HND)
/* iterative jusqu'à la fin de fichier */
DO WHILE I<=NBR
  ligne = READ(HND,'LINE')
  I = I + 1
  IF POS(«>D&eacute;tail&eacute;e</a>»,ligne) > 0 THEN DO
    ligne = READ(HND,'LINE')
    I = I + 1
    IF ENTREE == «ENTETE» THEN
      SAY RIGHT(ligne,LENGTH(ligne)-3)
  END
  IF POS(«<th nowrap>Plus d'info</th>»,ligne) > 0 THEN DO
    ligne = READ(HND,'LINE')
    ligne = READ(HND,'LINE')
    ligne = READ(HND,'LINE')
    I = I + 3
    NPOSITION = POS('=t> ',ligne)+5
    CODE = SUBSTR(ligne,NPOSITION,POS('</a>',ligne)-NPOSITION)
    ligne = READ(HND,'LINE')
    I = I + 1
    NPOSITION = POS('<small>',ligne)+7
    NOM = SUBSTR(ligne,NPOSITION,POS('</small>',ligne)-NPOSITION)
    ligne = READ(HND,'LINE')
    I = I + 1
    NPOSITION = POS('<small>',ligne)+7
    BOURSE = SUBSTR(ligne,NPOSITION,POS('</small>',ligne)-NPOSITION)
    ligne = READ(HND,'LINE')
    I = I + 1
    NPOSITION = POS('center>',ligne)+7
    DATECOURS = SUBSTR(ligne,NPOSITION,POS('</td>',ligne)-NPOSITION)
    ligne = READ(HND,'LINE')
    I = I + 1
    NPOSITION = POS('alt=E>',ligne)+6
    VALEURCOURS = SUBSTR(ligne,NPOSITION,POS('</b></td>',ligne)-NPOSITION)
    IF ENTREE <> «ENTETE» THEN
      SAY CODE NOM BOURSE DATECOURS VALEURCOURS
  END
END
```



SOMMAIRE CD

Numéro 27 • Décembre 2000

> Pour utiliser le CD-ROM, cliquez sur le fichier *index.htm* figurant sur la racine du CD. Chaque logiciel est accompagné d'un texte de présentation, une image ainsi qu'un bouton de lancement ou d'installation. Cliquez sur l'un de ces boutons pour installer ou lancer le programme.

Note : pour le programme *Jbuilder*, il est recommandé de lire le fichier fourni par Borland en cliquant sur le lien « à lire absolument ».



Sur votre CD-Rom



PROGRAMMES

● JBuilder 4 Foundation d'Inprise

(version d'évaluation en français, 30 jours)
JBuilder™ 4 est l'ensemble le plus complet d'outils visuels de développement pour la création de solutions Pur Java de bout en bout pour la plate-forme Java® 2 Enterprise. Développez et déployez rapidement des applications e-business J2EE™ pour de multiples serveurs d'applications incluant WebLogic® et Inprise™ Application Server™. Nouvelle version!

● SDK Microsoft.net

(version bêta, en anglais)
Cette première version Bêta contient tout ce qui est nécessaire pour écrire, compiler et tester des applications pour la future plate-forme Microsoft .Net. ASP+, le *Common Language Runtime*, la documentation, des exemples, des outils ainsi que les compilateurs en ligne. Nouveauté!

● Aide Pro 4.0 de Software Technology Resources

(version d'évaluation, en français)
AidePro4 est un environnement complet de développement spécialisé dans la création de documentations en ligne. Il vous permet, à partir d'un même projet, de créer un document au format HLP, au nouveau format HTML ou en version écrite.

LA COMPILATION S.T.R.

Les meilleurs outils du marché pour mettre à jour votre boîte à outils :

● Active Input 1.0

Ensemble de 4 contrôles de saisie liables à vos données qui vont donner à vos applications les qualités d'interactivité de Microsoft Internet Explorer.

● ActiveListBar 1.02

Donnez à vos applications le look Microsoft Office.

● ActiveThread Plus 3.01

Ensemble de 11 contrôles permettant de réaliser des interfaces personnalisées.

● ActiveToolBars Plus 2.01

Créez des barres d'outils et des barres de menus au standard Office 2000.

● ActiveTreeView 1.02

Pour une représentation hiérarchique de toutes vos données.

● Apollo Database Server 1.0

Une base de données client serveur aux grandes performances et à un coût réduit.

● Apollo VCL 1.0

Le package multi-primé qui permet de remplacer le BDE dans Delphi et dans C++ Builder.

● Basic Install 4.0

Outil de génération d'installation simple et rapide pour les développeurs.

● Net Tracker 4.0

Analyse du trafic sur votre site Internet, votre Proxy et vos fichiers de *logging* (firewall).

● Oletools 7.0

56 contrôles OCX personnalisés (OCX 32 Bits) pour satisfaire tous vos besoins de programmation.

● Ultragrid 1.0

Une nouvelle génération de grille optimisée pour la technologie OLE DB. Visual UML 2.5 : outil de modélisation d'objets pour Visual Basic, Visual FoxPro, VBA, VC++, Visual J++, etc.

REPertoire de la Redaction :

Retrouvez les sources des articles dans le dossier REDAC à partir de la racine du CD. Vous y trouverez notamment, les sources des articles du n°27 de « *Programmez !* ».

Programmez ! Le 1^{er} magazine de tous les langages • Supplément interactif N°27

JBuilder4

Nouvelle version

Système ouvert pour le développement d'applications 100% compatibles Java sous environnement Windows, Linux, et Solaris.

Borland JBuilder 4

VERSION D'ÉVALUATION EN ANGLAIS, 60 JOURS

Le Développement d'Applications Distribuées Pur Java

Borland JBuilder 4

Par Java. Développement Visual. J2EE

Environnement de développement dédié à la création de documentations en ligne. (version d'évaluation, en français)

Aide Pro 4.0 de Software Technology Resources

Kit de développement pour écrire, compiler et tester des applications pour la future plate-forme Microsoft .NET. (version bêta, en anglais)

Microsoft .net

STR S.T.R. : La compilation des meilleurs utilitaires de programmation du marché.

Dans C la Java


 Sur votre CD-Rom

Réutiliser du code existant et accélérer les applications Java est possible grâce à la Java Native Interface qui permet d'appeler du code natif. Si la sécurité s'en trouve affaiblie, le mariage du C/C++ et de Java promet du beau, du bon, du bon Net.

Frédéric Mazué
frederic.mazue@wanadoo.fr



Il y a de nombreuses raisons pour choisir Java pour l'écriture d'une application. Sa portabilité tout d'abord, ensuite Java offre de sympa-

thiques facilités pour la programmation réseau et l'accès aux bases de données (JDBC). Enfin Java est fortement sécurisé. Pour ces raisons Java est un langage de choix pour les entreprises disposant de serveurs de bases de données sur des réseaux hétérogènes. Deux ombres, néanmoins, ternissent ce tableau. Des milliers de lignes de code ont pu être écrites avant même l'existence de Java. Réécrire du C/C++, surtout du code C en Java n'est pas forcément une partie de plaisir. Cela peut être coûteux en temps et peut contraindre à écrire du code assez lourd, chose fréquente avec Java. Enfin Java souffre d'une lenteur d'exécution endémique.

Aborder Java autrement

Si l'on ne craint pas de faire une petite entorse à la sécurité, on pourra considérer comme positive l'utilisation de la

makefile pour Linux

```
OBJS = modulejava.o
COMPILE = g++ -I/usr/local/jdk1.2.2/include/ -I/usr/local/jdk1.2.2/include/linux/ -c -Wall
LINK = ld -shared -L/usr/lib/gcc-lib/i586-mandrake-linux/2.95.2/ -lstdc++

modulejava.so: $(OBJS)
    $(LINK) $(OBJS) -o modulejava.so
    cp -f modulejava.so /usr/local/jdk1.2.2/lib/perso/libmodulejava.so

%.o: %.cpp
    $(COMPILE) $< -o $@

clean:
    rm -f *.o
    rm -f *.so
```

JNI de Java. Comme son nom l'indique (*Java Native Interface*), elle permet d'appeler du code natif. Elle permet de concevoir l'écriture d'applications Java selon un autre angle : réutiliser du code C/C++ existant tout en offrant à ce dernier une interface graphique utilisateur homogène grâce à Swing.

Cette approche a plusieurs avantages :

- Réutiliser du code existant, *a priori* éprouvé et stable, ne coûte rien.
- Interfacer du code C/C++ avec Java est très facile, et donc peu onéreux, alors que la réécriture en Java de code non objet, du C par exemple, peut facilement dévorer un budget élyséen pour finalement aboutir à un résultat d'une lenteur inacceptable.
- Du code C/C++ bien écrit et sans interface graphique est normalement très facilement portable. Marier ce code avec une interface graphique rapidement écrite en Java grâce aux EDI du marché permet d'obtenir des applications portables à peu de frais.
- Conserver les facilités réseaux offertes par Java.
- Remédier à la lenteur de Java.
- Profiter de la puissance de Swing et éviter d'avoir à connaître de nombreux cadres GUI, chose fréquente dans un environnement C++.

En résumé cette approche permet de bénéficier des qualités intrinsèques à chacun des deux langages.

En revanche, la sécurité se trouve significativement amputée. La classe `SecurityManager` ne sera plus une barrière aussi efficace, et il est certain que du code natif défectueux peut déstabiliser des systèmes « délicats » comme Windows. Par contre, si la question de la sécurité n'est pas rédhibitoire, l'union de C/C++ et de Java peut donner de beaux enfants.

L'interface JNI

La JNI est bien conçue, facile et souple à utiliser ; ses possibilités sont complètes. Elle offre une interface pointeur pour accéder à ses méthodes. Il s'agit d'un pointeur sur des pointeurs de fonctions (voir la figure 1). Les pointeurs de fonctions (il s'agit ici de code natif) sont rangés dans un tableau qui rappelle à s'y méprendre une table de fonction virtuelle d'une classe C++. Ça ressemble encore plus à une interface COM à la sauce Microsoft. En fait, le principe de la JNI correspond si bien à COM que Sun envisage de supporter celle-ci dans des JDK futures dans le cas où elle viendrait à être une architecture multi-plates-formes.

Déclarer une méthode native

Rien n'est plus simple que de déclarer une méthode native en Java. Supposons que nous voulions écrire une méthode qui renvoie le double de la valeur qu'elle reçoit en paramètre. En Java, cela s'écrit comme ceci :

```
class maclasse
{
    public double doubleJava(double valeur)
    {
        return 2*valeur;
    }
}
```

La méthode doubleJava est écrite en Java, donc, le corps de la méthode est présent. Si nous voulons transformer cette méthode en une méthode native que l'on baptisera **doubleNative**, il suffit de supprimer le corps de la fonction et d'ajouter le mot clé « native ». Enfin la déclaration de la méthode sera suivie de « ; » comme ceci :

```
class maclasse
{
    public native doubleNative(double valeur);
}
```

Le corps de la méthode ne se situe donc plus dans la classe. Il doit faire partie d'une bibliothèque partagée qui doit être

chargée avant l'utilisation de la classe. Lors de ce chargement une édition de liens dynamiques est effectuée. Pour être tout-à-fait rigoureux, il convient de dire que le code natif peut aussi être placé dans une bibliothèque statique. Sun a conservé cette possibilité pour les systèmes ne supportant pas les bibliothèques partagées. Nous ne traiterons pas de ce cas particulier dans cet article.

Le moyen le plus simple pour charger une bibliothèque native consiste à placer le code de chargement dans un bloc « static » dans la classe. Un bloc « static » étant exécuté au premier chargement de la classe, nous sommes ainsi assurés que l'édition de liens a eu lieu avant que tout appel de méthode. Exemple :

```
public class maclasse
{
    public native double doubleNative(double valeur);

    static
    {
        System.loadLibrary(«modulejava»);
    }

    // code java 'normal' ici ...
}
```

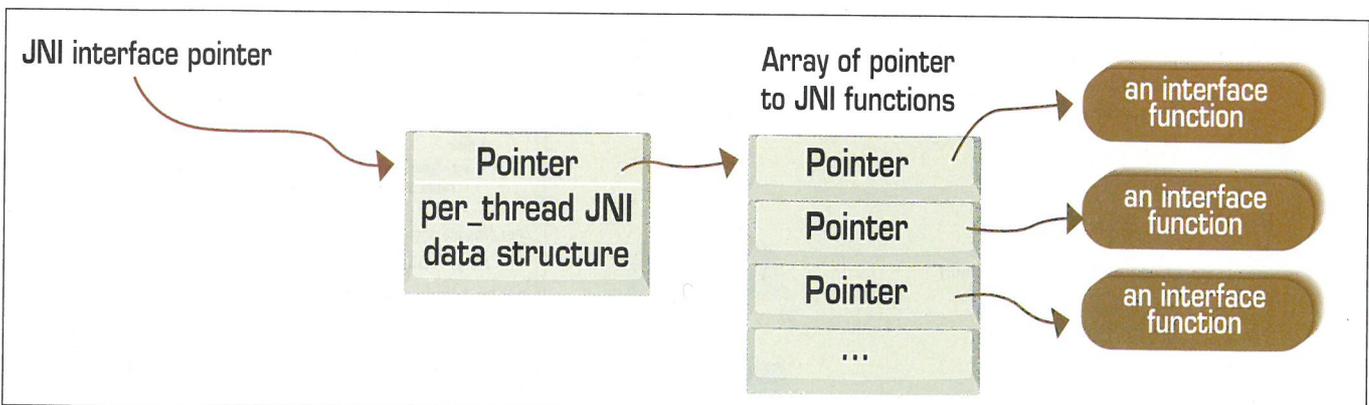
Le choix du nom de la bibliothèque est du ressort du programmeur et NE CORRESPOND PAS au nom de fichier de la bibliothèque. Le nom du fichier doit être tel que le système d'exploitation (et non la machine virtuelle) soit capable de charger la bibliothèque. Dans notre exemple, nous nommons notre bibliothèque « modulejava », cela signifie que :

- Sous Windows le fichier s'appellera « modulejava.dll »
- Sous Linux le fichier s'appellera « libmodulejava.so »

// classe d'exception Java personnalisée

```
public class ZeroDivision extends Exception
{
    public ZeroDivision(){}
    public ZeroDivision(String texte)
    {
        super(texte);
    }
}
```

2



> Figure 1: l'organisation d'une interface JNI (image extraite de la JDK 1.2 de Sun).

Dans sa documentation de la JDK 1.2.2, Sun déclare qu'un nom de fichier de bibliothèque en accord avec les règles du système d'exploitation cible et ledit fichier déposé dans un répertoire de bibliothèques du système sont des conditions suffisantes pour assurer le chargement de la bibliothèque. En fait, l'expérience montre que ce n'est pas entièrement exact. Sous Linux, par exemple, les répertoires contenant des bibliothèques partagées sont énumérés dans le fichier de configuration `ld.conf`. On pourrait penser que placer la bibliothèque dans un de ces répertoires suffit pour en assurer le chargement. Il en va différemment. Avec ma JDK 1.2.2,

il faut que le répertoire contenant la bibliothèque soit identifié dans une variable d'environnement exportée et nommée `LD_LIBRARY_PATH`. Sous Windows, il faut que le répertoire contenant la bibliothèque soit identifié dans la variable d'environnement `PATH`. Lorsque la machine virtuelle s'avère incapable de trouver une bibliothèque, elle lève une exception `java.lang.UnsatisfiedLinkError`.

Il n'est pas nécessaire d'écrire une bibliothèque par classe contenant des méthodes natives. Au contraire il peut être beaucoup plus rationnel de grouper toutes les méthodes natives devant être appelées par un programme dans une seule bibliothèque partagée. Dans ce cas, Sun précise que toutes les classes doivent être chargées par le même `ClassLoader`.

// Programme de démonstration — partie Java

```
public class maclasse
{
    private native void setFieldNative();
    private native String grandTexte(String texte);
    private native double doubleNative(double valeur);
    public native void lance_exception() throws ClassNotFoundException;
    public native double division(double dividende, double diviseur);

    static
    {
        System.loadLibrary("modulejava");
    }

    public static void main(String[] args)
    {
        double resultat;
        maclasse mcl = new maclasse();

        System.out.println("field avant modification: « + mcl.field");
        mcl.setFieldNative();
        System.out.println("field apres modification: « + mcl.field");

        System.out.println(mcl.grandTexte("un texte"));

        resultat = mcl.doubleNative(10);
        System.out.println(resultat);
        try
        {
            mcl.lance_exception();
        }
        catch(ClassNotFoundException cnfe)
        {
            System.out.println("ClassNotFoundException capturée ...");
        }
        resultat = mcl.division(5.0, 2.0);
        System.out.println(resultat);
        resultat = mcl.division(5.0, 0.0);
        System.out.println(resultat);
    }

    private double doubleJava(double valeur)
    {
        return valeur*2.0;
    }

    private int field = 0;
}
```

3

Notre exemple

Écrit pour Linux, l'adaptation de notre exemple pour Solaris va de soi. Pour Windows elle ne présente pas de difficulté à condition de bien respecter certains points particuliers :

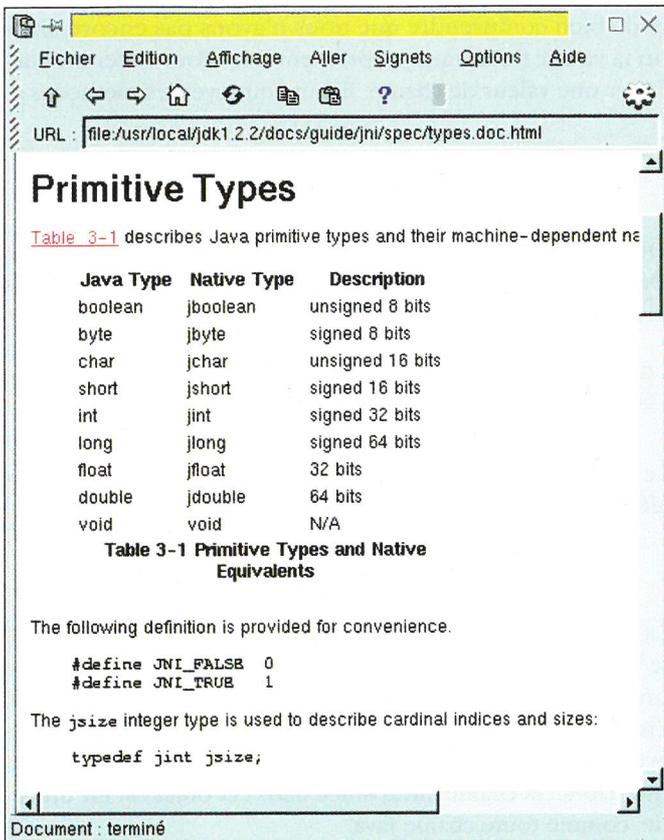
- Le compilateur doit exporter tous les noms de méthodes natives, faute de quoi le mécanisme de liaison dynamique échouerait et la machine virtuelle lèverait l'exception `java.lang.UnsatisfiedLinkError`. Consultez la documentation de votre compilateur relative à l'exportation des noms de fonctions d'une dll.
- Il faut veiller à ce que le compilateur ne décore pas le nom des fonctions exportées derrière votre dos. Ceci est une spécialité de Visual C++. Reportez-vous à sa documentation.

The screenshot shows a web browser window with the URL `file:/usr/local/jdk1.2.2/docs/guide/jni/spec/types.doc.html`. The page title is "Type Signatures". The content includes a table mapping JNI type signatures to Java types:

Type Signature	Java Type
Z	boolean
B	byte
C	char
S	short
I	int
J	long
F	float
D	double
L <i>fully-qualified-class</i>	fully-qualified-class
[<i>type</i>]	<i>type</i> []
(<i>arg-types</i>) <i>ret-type</i>	method type

Below the table, it says "Table 3-2 Java VM Type Signatures". An example is given: "For example, the Java method: `long f (int n, String s, int[] arr);` has the following type signature: `(ILjava/lang/String; [I)J`".

Écran 2: les éléments constitutifs d'une signature de méthode en Java.



> Écran 3: la JVM convertit les paramètres Java en types primitifs lors des appels de méthode natives.

Dans notre exemple, nous déclarons cinq méthodes natives afin d'illustrer les utilisations les plus courantes. Les méthodes privées et publiques cohabitent dans un beau désordre. En fait, les attributs `private` ou `public` n'ont pas de conséquence sur l'écriture du code natif:

```
public class maclasse
{
    private native void setFieldNative();
    private native String grandTexte(String texte);
    private native double doubleNative(double valeur);
    public native void lance_exception() throws ClassNotFoundException;
    public native double division(double dividende, double diviseur);

    // etc...
}
```

- `setFieldNative` montre comment modifier une variable de classe depuis une méthode native.
- `grandTexte` illustre le travail «à la chaîne». La méthode reçoit un objet «String» et en retourne un autre.
- `doubleNative`. Nous en avons déjà parlé. Mais, afin de varier les plaisirs nous verrons comment appeler une méthode de classe Java depuis une méthode native.
- `lance_exception` illustre la manière de lancer une exception java depuis du code natif.
- `division` montre comment lancer une exception personnalisée depuis du code natif. Lorsqu'il travaille avec des

doubles, Java accepte de diviser par 0.0 et retourne la valeur «Infinity». Jugeant cette valeur peu pratique à manipuler, nous préférons lancer une exception.

Savoir s'entêter

Une fois la classe écrite en Java, il faut s'occuper du code natif. La JDK vient avec un compilateur nommé «javah» qui génère des en-têtes C/C++ à partir de code Java. La première opération consiste donc à compiler le code Java avec lui: `javah maclasse` (et non `javah maclasse.java`)

Ce compilateur génère un fichier en-tête `.h` directement utilisable. Jeter un coup d'œil dans ce fichier s'avère instructif. D'abord on y voit que tous les noms de méthodes natives ont été décorés selon le schéma immuable:

```
Java_classe_méthode
```

On voit également que les méthodes ont des «signatures». Une signature de méthode est une petite chaîne de caractères décrivant les types des paramètres reçus et le type de la valeur renvoyée par la méthode. Comprendre les signatures devient utile lorsqu'il s'agit d'accéder à une variable ou une méthode de classe Java. La JDK donne toutes les informations concernant les signatures (voir l'écran 2).

Sont enfin dignes d'intérêt les paramètres que la JVM envoie aux méthodes natives. Les types Java sont automatiquement convertis en type natifs dits primitifs. Aucune difficulté dans cette partie comme on peut le voir sur l'écran 3.

Deux autres paramètres méritent toute notre attention:

- `object obj` est un pointeur sur la classe Java propriétaire de la méthode native appelée. Disons pour simplifier que c'est la même chose qu'un pointeur `this`.
- `JNIEnv *env` est un pointeur sur la fameuse interface JNI. Ceux qui connaissent bien Windows peuvent y voir une analogie avec un pointeur sur une interface COM comme nous l'avons dit au début de l'article. Ce pointeur sert à appeler les très nombreuses méthodes de l'interface. Ces méthodes permettent de tout faire: par exemple, trouver une classe, modifier une valeur de champ, etc. Rien de plus simple que d'appeler une méthode de la JNI depuis C++:

```
env->Method JNI();
```

Depuis C la syntaxe est un peu plus lourde:

```
(*env)->Method JNI();
```

Nous allons maintenant, notre programme d'exemple à la main, passer en revue quelques possibilités de la JNI.

Modifier une variable de classe

Notre classe d'exemple comporte un champ privé nommé `field`. Grâce à la JNI, nous pouvons accéder à ce champ et le modifier à notre gré. Que le champ soit `private` ou `public` n'a pas de répercussion sur la façon d'y accéder avec la JNI. Pour

// fichier en-tête généré par javah

```

/* DO NOT EDIT THIS FILE - it is machine generated */
#include <jni.h>
/* Header for class maclasse */

#ifdef _Included_maclasse
#define _Included_maclasse
#ifdef __cplusplus
extern «C» {
#endif
/*
 * Class:   maclasse
 * Method:  division
 * Signature: (DD)D
 */
JNIEXPORT jdouble JNICALL Java_maclasse_division
    (JNIEnv *, jobject, jdouble, jdouble);

/*
 * Class:   maclasse
 * Method:  doubleNative
 * Signature: (D)D
 */
JNIEXPORT jdouble JNICALL Java_maclasse_doubleNative
    (JNIEnv *, jobject, jdouble);

/*
 * Class:   maclasse
 * Method:  grandTexte
 * Signature: (Ljava/lang/String;)Ljava/lang/String;
 */
JNIEXPORT jstring JNICALL Java_maclasse_grandTexte
    (JNIEnv *, jobject, jstring);

/*
 * Class:   maclasse
 * Method:  lance_exception
 * Signature: ()V
 */
JNIEXPORT void JNICALL Java_maclasse_lance_1exception
    (JNIEnv *, jobject);

/*
 * Class:   maclasse
 * Method:  setFieldNative
 * Signature: ()V
 */
JNIEXPORT void JNICALL Java_maclasse_setFieldNative
    (JNIEnv *, jobject);

#ifdef __cplusplus
}
#endif
#endif

```

4

pouvoir accéder à un champ, il faut savoir à quelle classe nous avons affaire. Ceci se fait avec la méthode `GetObjectClass` qui retrouve la définition de la classe à partir du pointeur « this ». Ensuite il faut récupérer le champ grâce à la méthode `GetFieldID`. On remarquera qu'entre autres paramètres, cette méthode attend une signature. En effet, tout comme les méthodes de classes, les valeurs de classes ont une signature constituée, d'ailleurs, de la même façon. À ce stade, nous avons accès au champ seulement. Il impor-

te de bien comprendre que nous n'avons pas encore obtenu la valeur du champ proprement dite. Pour obtenir ou affecter une valeur de champ, il faut employer les méthodes :

```

Get<type>Field()
Set<type>Field()

```

Où `<type>` correspond au type du champ en question. Dans notre exemple, le champ contient un entier, donc les méthodes à employer sont :

```

GetIntField()
SetIntField()

```

Le programme d'exemple utilise la deuxième méthode pour affecter la valeur 1 au champ `field`.

Travail à la chaîne

La méthode `grandTexte` du programme d'exemple illustre le travail avec les chaînes Java. Depuis la classe Java on passe une « String » à la fonction native. Celle-ci reçoit alors une « jstring » qui reste une chaîne Java. Cela veut dire que nous avons cette fois affaire à un objet, et non à une valeur primitive. Donc, la chaîne encapsulée dans cet objet est en Unicode, comme toute chaîne Java.

Pour les besoins de la démonstration nous supposons que notre compilateur C++ ne connaît pas les « wstring » de la bibliothèque standard (c'est d'ailleurs souvent le cas, hélas). Nous devons donc récupérer la chaîne de caractères sous la forme ASCII traditionnelle, la modifier, puis la transformer à nouveau en une chaîne Unicode avant de la retourner à la fonction appelante dans le programme Java.

Si vous examinez le programme d'exemple, vous verrez que ces opérations sont bien effectuées. Vous y verrez aussi une chose supplémentaire très importante. Globalement, le programmeur qui travaille avec la JNI n'a pas davantage à se préoccuper de la gestion de la mémoire des objets Java que de Java lui-même. Le ramasse-miettes s'occupe (normalement) de tout. Il y a quand même quelques exceptions auxquelles il faut être attentif. Lorsque nous appelons la méthode `GetStringUTFChars`, cela a pour effet d'interdire au ramasse-miettes de Java de disposer de la mémoire occupée par l'objet chaîne. Il faut qu'il en soit ainsi, sinon nous ne pourrions pas nous fier au pointeur de caractères retourné par la méthode. En contrepartie, pour éviter une fuite mémoire, nous devons dans ce cas nous préoccuper de la libération de l'objet. Cela se fait au moyen de la méthode `ReleaseStringUTFChars`.

Appeler du code Java depuis du code natif

Il est très facile d'appeler une méthode Java de notre classe. Appeler une méthode publique Java d'une autre classe ne l'est guère moins. Ce qui a été dit à propos des valeurs de champ reste valable. Il faut d'abord récupérer une classe,

puis la méthode voulue grâce à son nom et à sa signature. Pour appeler réellement la méthode, la JNI fournit une panoplie complète de méthodes `Call<type>Method` où `<type>` est le type de la valeur de retour de la méthode de classe appelée. Si la méthode est «static», la JNI met d'autres méthodes à notre disposition. Cette remarque s'applique également aux champs statiques.

// programme de démonstration — partie C++

```
#include <string>
#include <jni.h>
#include «maclasse.h»

using namespace std;

JNIEXPORT jdouble JNICALL Java_maclasse_division
(JNIEnv *env, jobject obj, jdouble dividende, jdouble diviseur)
{
    if(diviseur == 0.0)
    {
        jclass ZeroDivision = env->FindClass(«ZeroDivision»);
        env->ThrowNew(ZeroDivision, «Vous avez fait une division par zero»);
    }
    return dividende/diviseur;
}

JNIEXPORT void JNICALL Java_maclasse_lance_1exception(JNIEnv *env, jobject obj)
{
    jclass nfe = env->FindClass(«java/lang/ClassNotFoundException»);
    env->ThrowNew(nfe, «Ce texte n'apparaîtra pas»);
}

JNIEXPORT jdouble JNICALL Java_maclasse_doubleNative(JNIEnv *env, jobject obj, jdouble valeur)
{
    jdouble resultat;
    jmethodID jmid;

    jmid = env->GetMethodID(env->GetObjectClass(obj), «doubleJava», «(D)D»);
    resultat = env->CallDoubleMethod(obj, jmid, valeur);
    return resultat;
}

JNIEXPORT void JNICALL Java_maclasse_setFieldNative(JNIEnv *env, jobject obj)
{
    jfieldID jfid;

    jfid = env->GetFieldID(env->GetObjectClass(obj), «field», «I»);
    env->SetIntField(obj, jfid, 1);
}

JNIEXPORT jstring JNICALL Java_maclasse_grandTexte(JNIEnv *env, jobject obj, jstring str)
{
    const char *pchar;
    jstring ls;

    pchar = env->GetStringUTFChars(str, NULL);
    string s(pchar);
    s += « plus long »;
    ls = env->NewStringUTF(s.c_str());

    env->ReleaseStringUTFChars(str, pchar);
    return ls;
}
```

5

Lever une exception

Du code natif bien écrit doit lever des exceptions Java en cas de problème afin que la JVM ne perde pas les pédales et que sa stabilité ne soit pas compromise.

Il y a deux cas de figure. Soit Java dispose déjà d'une exception qui nous convient, soit nous devons écrire nous-même

une classe exception personnalisée qui puisse être lancée depuis du code natif ; pour ce faire, il faut et il suffit de dériver dans du code Java une classe de Throwable. Le programme d'exemple dérive une classe de la classe `Exception`, pour traiter le cas de la division par zéro. Il ne s'agit ici que d'un exemple. Dans un programme réel, il sera opportun de rechercher la classe la plus adaptée pour servir de base à notre exception personnalisée. Ici cela pourrait être `RuntimeException` ou `ArithmeticException`, par exemple.

Dans un cas de figure comme dans l'autre, lever une exception est un jeu d'enfant. D'abord il faut trouver la classe exception que l'on veut lever au moyen de `FindClass`, puis lever l'exception avec `ThrowNew`.

Le programme d'exemple capture une exception Java, puis s'arrête sur notre exception personnalisée non capturée. Les deux exceptions sont lancées depuis le code natif.

Conclusion

Il y a encore beaucoup à dire sur la JNI tant celle-ci est riche. Par exemple nous n'avons pas parlé du tout du travail avec les tableaux. Malgré cela, si les principes de base exposés dans cet article sont bien compris, il est facile d'extrapoler et de devenir un expert de la JNI. Celle-ci est un moyen intéressant pour écrire des applications portables présentant des interfaces utilisateurs homogènes dans des environnements hétérogènes. La JNI permet de réutiliser du code C/C++ existant.

Enfin, on pourra aussi s'intéresser à la JNI dans le seul but d'optimiser la rapidité d'exécution d'un programme Java en ré-écrivant en C/C++ les parties de codes les plus gourmandes en temps de calcul. Notons également que le site américain de Sun propose de très nombreux liens vers des sociétés tierces qui proposent des produits étendant et simplifiant la mise en œuvre de la JNI, c'est par exemple, le cas de `JunC++ion` de la société `Codemesh` pour la plate-forme `Solaris`. ■

<http://solutions.sun.com>

Commandez votre déjeuner via le Web



Vous ne connaissez rien au langage Python ? Aucune importance, l'auteur démontre combien il est facile d'écrire avec ce langage, de créer une application en quelques minutes et par là, d'être servi sans se retrouver chocolat.

Par Cheng-Chai Ang, *Linux Journal*
Traduction de Frédéric Mazué

Peut-être aurais-je dû intituler cet article « Comment j'ai découvert Python et jeté tout le reste ». J'ai toujours voulu écrire des applications web sans savoir comment m'y prendre. Se lancer là-dedans était assez intimidant. Après avoir remis pendant des années mes débuts au lendemain, je me suis finalement retrouvé à écrire – pour ma société – une application devant récupérer des informations depuis une page web. Ces valeurs sont envoyées à un script CGI, validées dans une base de données SQL, distribuées à un processus en attente *via* des *sockets* pour, en bout de course, retourner un résultat sur la page web.

Par chance, je suis tombé sur un langage de script appelé Python. J'ai lu une interview d'Éric Raymond (l'auteur de *The Cathedral and the Bazaar*, éd. O'Reilly, ndlr), dans le numéro de décembre 1999 de *Linux Journal*. Il y déclarait ne programmer désormais qu'en Python. À cet époque j'essayais d'implémenter le système décrit plus haut en Perl et ne m'en sortais pas. Si Python suscitait la considération d'Éric, cela valait certainement le coup de l'essayer. J'ai achevé ma tâche en seulement un peu plus de deux heures, et ceci en utilisant un langage dont je n'avais pas entendu parler auparavant. Au risque de perdre un avantage professionnel, je voudrais partager avec d'autres les facilités offertes par Python, spécialement pour la programmation CGI (et pour à peu

près tout le reste d'ailleurs). Comme les explications données plus haut peuvent paraître trop techniques ou révéler des informations sur un projet réel en cours, j'ai choisi de travailler, ici, avec un exercice plus intéressant et moins confidentiel.

La description du problème

Mon lieu de travail se trouve dans un quartier un peu isolé (culinairement parlant, à l'exception de la cantine où la chère est bonne, mais... chère). Mes collègues et moi-même devions organiser un roulement pour désigner qui descendrait les trois étages et partirait au loin en quête de nourriture. Un système de commande par le biais du Web me paraissait une solution beaucoup plus satisfaisante, mais comme je n'avais pas fait la moindre programmation CGI auparavant, la besogne me paraissait accablante. En fait, écrire des systèmes CGI web peut être vraiment simple, spécialement si l'on utilise Python. (OK, les gourous Perl peuvent désapprouver, mais c'est là tout l'enjeu : un gourou Perl contre un novice Python).

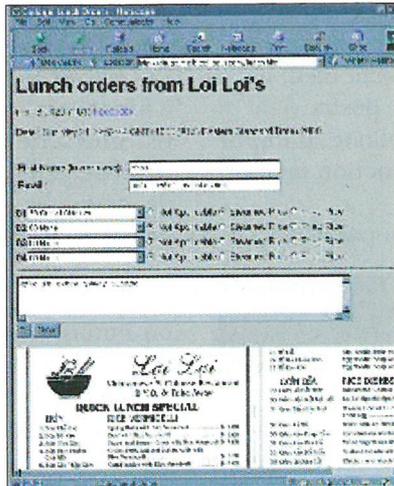
Je savais à peu près comment je voulais structurer mon système. Il y aurait une page web avec une liste déroulante contenant le menu du restaurant et en cliquant sur le bouton

« **submit** » un e-mail serait envoyé à un fournisseur de déjeuners. En me fiant aux ouï-dire et à une recherche rapide sur le Net, j'ai choisi d'utiliser les outils suivants :

Ressources *Learning Python*, M. Lutz et D. Ascher, O'Reilly (1999).
Manuels de référence Python : <http://www.python.org/>
Javascript, Laura Lemay, SAMS Publishing (1997).

- JavaScript pour le frontal client (la page web).
- Python pour le côté serveur.
- Apache pour le serveur web, qui est distribué avec Linux (en tout cas, je l'ai trouvé sur ma Red Hat 6.2). Il existe aussi une version Windows si quelqu'un préfère.
- Linux pour le système d'exploitation du serveur.

J'ai rapidement créé mes pages web. J'ai décidé de faire dans la simplicité : une liste déroulante et quelques boutons radio (voir l'écran 1). J'aurais pu employer un éditeur HTML, mais je me suis dit que cela ne valait pas le coup d'apprendre à utiliser un nouveau logiciel. Alors j'ai tout fait à la main. Ce que je voulais n'était pas si compliqué. Il fut facile d'installer le serveur web sur Linux. Quand j'ai installé Linux, l'option pour l'installation automatique d'Apache était cochée. Quand j'ai tapé l'URL *localhost* sous Netscape celui-ci a affiché une page comportant un message disant que si je voyais cette page tout était OK! Wahou... Si facile! (Voir <http://www.apache.org> pour plus de détails). Vous devrez probablement être *root* (le superutilisateur) pour faire cette installation.



> Écran 1 : le menu de commande des déjeuners.

Utiliser Javascript pour écrire une page web semble assez convenable. Il y a plusieurs fonctions pour la vérification des données en entrées (`ValidLength()` et `ValidEmail()`). `MenuHeader()` affiche la partie en-tête de la page. Chaque appel à `MenuEntry()` affiche une ligne d'entrée. Dans notre cas, elle est appelée quatre fois, une fois pour chaque composant de notre déjeuner (voir le listing sur CD-Rom). Les lignes les plus délicates concernent le traitement du bouton `submit` :

```
<FORM NAME="order" onSubmit="return Validate();">
ACTION="http://localhost/cgi-bin/lunch.cgi"
METHOD="POST">
```

Il y a deux façons pour qu'une page web communique avec

```
#
#!/usr/bin/env python
# first.cgi

# importer le module cgi
import cgi

# nécessaire pour le navigateur
print «Content-Type: text/plain\n\n»

# récupérer les valeurs
lunchForm = cgi.FieldStorage()

for name in lunchForm.keys():
    print «Key= « + name + « Value= « \
        + lunchForm[name].value + « »

print «bye.»
#
```



un script CGI : `GET` et `POST`. En bref, `GET` envoie l'information sous forme d'une URL (en surfant sur Internet, vous avez dû voir des URL ressemblant à ceci :

`http://localhost/script.cgi?param1=value1¶m2=value2`)

Quand `POST` est employé, les informations du formulaire sont envoyées sur l'entrée standard. Cela signifie que le script CGI doit lire l'entrée standard et ensuite analyser l'entrée pour en séparer les différents paramètres. Il est généralement admis que `POST` est meilleur (plus robuste, pas limité par un nombre de caractères maximum). Les méthodes pour extraire les données diffèrent, selon que `GET` ou `POST` est employé, mais Python vous le cache (ce qui est bien).

Visualiser lunch.html via le serveur web

J'ai ensuite placé le fichier `lunch.html` dans le répertoire `/home/httpd.html` :

```
$ cp lunch.html /home/httpd/html
```

(Ceci est le répertoire par défaut où Apache recherche les fichiers html. Apache peut être configuré pour regarder à un autre endroit). Après avoir fait cela, vous pouvez voir à quoi ressemble `lunch.html` à l'adresse `http://localhost/lunch.html`, ceci aussi bien avec Netscape qu'avec un autre navigateur. Abordons l'écriture du script CGI avec Python. Python fait partie de la distribution Red Hat. (Voir <http://www.python.org>). Après avoir consulté la documentation de Python, documentation qui fait partie du langage, j'ai écrit mon premier script qui ressemblait au `listing 1`.

C'est essentiellement une version «cannibalisée» d'un exemple trouvé dans la documentation Python. Pour pouvoir utiliser ce script son nom doit être spécifié dans la partie `ACTION` du traitement du bouton `submit` dans la page html. Donc, changez le script spécifié à `ACTION` de `lunch.cgi` en `first.cgi`.

Tester l'installation

J'ai copié `first.cgi` dans le répertoire `/home/httpd/cgi-bin` :

```
$ cp first.cgi /home/httpd/cgi-bin
```

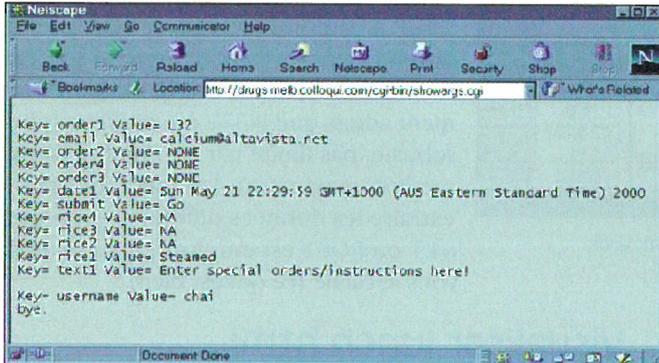
J'ai essentiellement interrogé toutes les variables envoyées au script par le formulaire et les ai affichées en écho. Ainsi toutes les sorties sont affichées par le navigateur. Quand l'utilisateur clique sur le bouton «Go» de la page `lunch.html` et que le script `first.cgi` est ainsi activé, les sorties retournées vers le navigateur ressemblent à ce que vous voyez à l'écran 2.

Vous noterez que les clés du script CGI correspondent aux variables que j'emploie dans `lunch.html`.

Après m'être assuré que ce script fonctionnait bien, je l'ai modifié pour qu'il soit conforme à mes désirs (voir le listing 2). Le code Python est direct et explicite. Il importe le module CGI, et appelle ensuite la fonction membre `FieldStorage()` de CGI.

Que l'information soit envoyée par `GET` ou par `POST`, cela

reste transparent pour vous. Voilà donc comment les informations envoyées par une page web sont récupérées. Ces informations peuvent alors être extraites en accédant à `lunchForm`. Le corps de mail envoyé est alors construit par une série de «*write*» à `sendmail` (programme UNIX de postage de mail). J'ai décidé de poster la commande de déjeuner à l'utilisateur `lunch@localhost/`. Un alias peut être ajouté dans le fichier `/etc/aliases` :



Écran 2 : le résultat de l'exécution du script `first.cgi`.

Listing 3. Listing `lunch.cgi`

```
#!/usr/bin/env python
import cgi
import os

print «Content-Type: text/plain\n\n»
print «Hello, lunch patron », TheForm[«username»].value
print «Dont forget to add 50 cents for delivery. »

print «\n\nThe following information was sent:»

lunchForm = cgi.FieldStorage()

p = os.popen(«%s +% SENDMAIL, «w»)
p.write(«To: lunch@drugs.melb.colloqui.com\n»)
p.write(«From: « + lunchForm[«username»].value + « <» \
    + lunchForm[«email»].value + «>\n»)
p.write(«Subject: loi loi\n»)
p.write(«\n») # blank line separating headers from body
p.write(«SourceIP « + os.getenv[«REMOTE_ADDR»] + «\n»)

p.write(lunchForm[«email»].value + «\n»)
p.write(lunchForm[«date1»].value + «\n»)
p.write(lunchForm[«username»].value + « wants\n»)
for i in ('1', '2', '3', '4'):
    orderNo = «order» + i
    riceType = «rice» + i
    print(i + «. « + lunchForm[orderNo].value + « with « + \
        lunchForm[riceType].value + « rice.»)
    p.write(i + «. « + lunchForm[orderNo].value + « with « + \
        lunchForm[riceType].value + « rice.\n»)

p.write(«\nSpecial Instructions: « + \
    lunchForm[«text1»].value + «\n»)

smStatus = p.close()
if smStatus != 0:
    print «Sendmail exit status=», smStatus

print «Finished!»
```



```
lunch:      chai@localhost
```

où l'utilisateur `chai@localhost` est celui qui organise les commandes. Ainsi, si l'organisateur vient à changer, il suffit seulement de modifier le fichier `/etc/aliases`. Il n'est pas nécessaire de modifier le script CGI (il faut lancer la commande `newaliases` pour que les changements dans `/etc/aliases` prennent effet). Facile non ? Cela aurait pu être bien pire.

Tout mettre ensemble

J'ai ensuite copié `lunch.cgi` dans le répertoire `/home/httpd/cgi-bin`. J'ai ouvert Netscape, tapé l'URL `http://localhost/lunch.html`, rempli le formulaire, sélectionné ma commande et cliqué sur «*Go*».

Peu après, j'ai reçu un e-mail décrivant ma commande. Voici l'aspect du mail de commande de déjeuner, tel qu'envoyé par le script CGI :

```
>From nobody@localhost Wed Apr 26 11:01:50 2000<\n>
Delivered-To: ccang@localhost
Date: Wed, 26 Apr 2000 11:01:48 +1000
To: lunch@localhost
From: chai <calcium@altavista.net>
Subject: loi loi

Sender: nobody@localhost

SourceIP 194.118.1.1
calcium@altavista.net
Wed Apr 26 11:01:01 GMT+1000 (EST) 2000
chai wants
1. 1.39 with Steamed rice.
2. NONE with NA rice.
3. NONE with NA rice.
4. NONE with NA rice.

Instruction spéciale : faites quelque chose de délicieux
```

L'écran 3 montre à quoi ressemble la page web une fois que la requête a été envoyée.

Conclusion

Est-ce qu'un système de commande en ligne vaut plus qu'une commande écrite sur un bout de papier ? On peut en débattre. En tous cas, c'est plus amusant (pour moi, tout du moins). On peut bien sûr apporter quelques amé-

Initiation à OpenGL 2^e partie

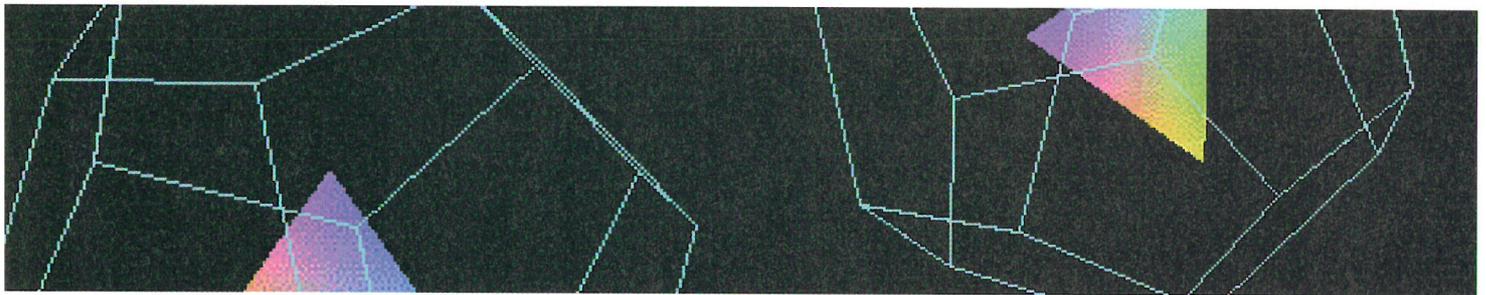
Nous entrons aujourd'hui dans le vif du sujet en apprenant à manipuler les éléments constitutifs d'une animation OpenGL.

Frédéric Mazué

fredericmazue@wanadoo.fr



Sur votre CD-Rom



Dans le numéro 25 de *Programmez!* (octobre 2000), nous avons appris à écrire, compiler et lire un programme C/C++/OpenGL sur les plates-formes Linux et Windows. Nous disposons maintenant d'un programme de base, une sorte de coquille, qui nous servira de point de départ pour tous les exemples de cette série d'articles.

Nous pouvons maintenant nous consacrer à la programmation d'OpenGL proprement dite.

Qu'est-ce qu'une image OpenGL ?

Que ce soit avec OpenGL ou avec une autre bibliothèque graphique 3D, une image est, la plupart du temps, la restitution d'un objet réel (ou sa représentation, si vous préférez), donc un objet occupant un volume dans l'espace physique. Le propos de toute bibliothèque graphique 3D est de donner sur un écran plat une illusion de volume. Pour cela, sont utilisées diverses techniques plus ou moins complexes telles que l'éclairage et les dégradés de couleurs. Mais à la base de tout il y a l'objet lui-même. Pour restituer un objet avec OpenGL, on « dessine » les surfaces enveloppantes des volumes élémentaires constituant l'objet. Les surfaces enveloppantes sont à leur tour découpées en surfaces géométriques élémentaires, c'est-à-dire en triangles ou en polygones. En fait, il faudrait dire « découpage en triangles seulement », car lorsque l'on demande à OpenGL de dessiner un polygone, elle découpe automatiquement ce dernier en triangles de façon interne. Quand un volume élémentaire ne présente pas de surface plane, par exemple une sphère ou

un tore, alors il faut faire une approximation de la surface gauche (une surface non plane est dite gauche) en la remplaçant par une quantité de petits triangles enveloppants. Prenons un exemple concret : la sphère. La sphère n'étant pas représentable, on la transforme en un icosaèdre, c'est-à-dire une surface constituée de triangles. Il est évident que plus le nombre de triangles est grand, plus la taille de ceux-ci est petite, et donc, la sphère moins approximative. Malheureusement, il est non moins évident que plus le nombre de triangles est élevé, plus le temps de calcul de l'image sera long. Tout le problème de ceux qui créent des environne-

L'hamburger OpenGL Comme pour toutes les règles à confirmer, il faut une exception. Celle du sandwich OpenGL concerne les listes d'affichage. Une liste d'affichage est un groupe de commandes stockées pour une utilisation ultérieure. On utilise les listes d'affichage pour le rendu d'images complexes. En raison du caractère fourre-tout de la liste d'affichage, la construction de celle-ci s'apparente plus au hamburger qu'au sandwich. Cette fois, les tranches de pain sont :

```
glNewList(.....)
// stockage des commandes de dessins
glEndList()
```

Les commandes stockées dans la liste entrent en action à l'appel de la fonction de `glCallList`, cette dernière fonction NE DEVANT PAS figurer dans un sandwich `glBegin glEnd`, faute de quoi OpenGL risquerait l'indigestion. :)

ments 3D est de trouver un équilibre judicieux entre la qualité du rendu et le temps de calcul. Restituer des environnements 3D complexes est une tâche ardue où l'expérience et l'habileté du programmeur sont irremplaçables.

Pour nous débutants, ce qu'il importe de retenir est qu'une image OpenGL est un «paquet» de triangles et de polygones. Certes, OpenGL peut aussi tracer de simples points et lignes à l'écran. Nous n'entrerons pas dans le détail de ces possibilités dans le cadre de cette série d'articles. Il nous suffit de dire que les principes généraux sont les mêmes pour les lignes et les points que pour les surfaces. Le lecteur intéressé se reportera à la documentation d'OpenGL.

Le sandwich OpenGL

Les surfaces ne sont pas non plus représentées directement par OpenGL. OpenGL travaille avec les vertices (un vertex, des vertices), c'est-à-dire les sommets des surfaces. Selon le mode choisi par le programmeur, OpenGL reliera ces sommets par des lignes (mode filaire) ou remplira la surface délimitée par ceux-ci (mode de remplissage). La définition des vertices constitue la partie «nourrissante» entre les tranches de pain du sandwich OpenGL que sont les fonctions `glBegin` et `glEnd()`.

Ainsi le code de définition d'une surface à l'allure suivante :

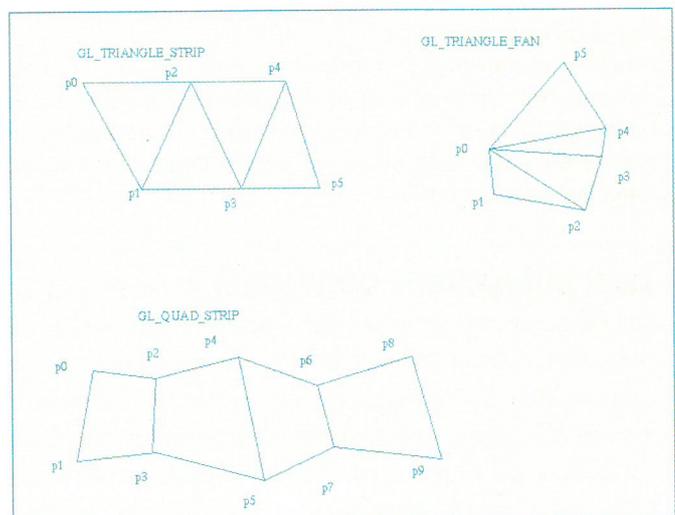
```
// commencer la définition d'un groupe de sommets
glBegin(GLenum mode)
    // définir ici les sommets
// fin de l'opération
glEnd()
```

`GLenum mode` définit comment seront utilisés les sommets par OpenGL et donc définit ce qui sera tracé à l'écran. Mode peut prendre les valeurs suivantes :

- `GL_POINTS` : chaque sommet est représenté par un point.
- `GL_LINES` trace des segments de droites entre les paires de sommets. Si les sommets sont en nombre impair, le dernier est ignoré.
- `GL_LINE_STRIP` trace des segments de droites reliés les uns aux autres. Il faut définir au moins deux sommets pour qu'un tracé ait effectivement lieu.
- `GL_LINE_LOOP` : comme ci-dessus, sauf que le dernier segment rejoint le premier sommet. Autrement dit, on trace un ensemble fermé de segments.
- `GL_TRIANGLES` dessine des triangles en utilisant les sommets par groupe de trois. Si le nombre de sommets n'est pas un multiple de trois, le ou les deux derniers sont ignorés.
- `GL_TRIANGLE_STRIP` dessine une série de triangles telle que les deux derniers sommets du triangle précédent sont les deux premiers sommets du triangle suivant (voir la **figure 1**).
- `GL_TRIANGLE_FAN` dessine une série de triangles telle que tous «pointent» vers le premier sommet (voir la **figure 1**).
- `GL_QUAD` dessine une série de quadrilatères. Si le nombre de sommets n'est pas un multiple de 4, le ou les derniers sommets sont ignorés.

- `GL_QUAD_STRIP` dessine une série de quadrilatères tels que le dernier côté du quadrilatère précédent est le premier côté du quadrilatère suivant (voir la **figure 1**).
- `GL_POLYGON` dessine un polygone avec les sommets définis. Pour sélectionner le mode de dessin du polygone parmi (entre autres) les modes point, filaire et remplissage, appeler la fonction `glPolygonMode`. Dans tous les cas OpenGL ne sait tracer que les polygones convexes (voir l'**encadré des polygones convexes**).

Note importante : l'appel à OpenGL ne provoque pas le dessin à l'écran. Il est heureux qu'il en soit ainsi car cela offre la possibilité de définir de nombreuses figures qui seront rendues ensuite toutes en même temps. L'exécution des commandes se fait à l'appel de la fonction `glFlush` ou comme dans le programme d'exemple lors de l'appel de la fonction `glutSwapBuffers` de la bibliothèque glut. Cette fonction de gestion de l'affichage en double tampon appelle à son tour `glFlush`.



> Figure 1: illustration des modes `GL_TRIANGLE_STRIP`, `GL_TRIANGLE_FAN` ET `GL_QUAD_STRIP`.

Arrêtons-nous là pour le tracé des figures géométriques. Le lecteur se doutera que nous sommes (très) loin d'avoir fait le tour de la question et approfondira ses connaissances grâce à la documentation d'OpenGL ou à des ouvrages spécialisés.

Les facilités offertes par glut

Prenons, en revanche, le temps de mentionner que la bibliothèque glut comporte 18 fonctions permettant le rendu rapide de 9 volumes préfabriqués qui sont la sphère, le cube, le tore, l'icosaèdre, l'octaèdre, le tétraèdre, le dodécaèdre, le cône et la théière (oui, vous avez bien lu, une théière).

Chacun de ces 9 volumes peut être rendu en mode filaire ou en mode de remplissage, ce qui nous donne 18 fonctions. Le préfixe `glutWire` indique un travail en mode filaire tandis que le préfixe `glutSolid` indique un travail en mode de remplissage.

➤ Ainsi, `glutWireSphere` dessinera une sphère en mode filaire, tandis que `glutSolidCone` dessinera un cône en mode remplissage.

Le programme d'exemple et l'animation glut.

Faisons maintenant une pause et voyons comment obtenir un effet d'animation avec glut. Nous reviendrons à OpenGL ensuite. Le programme d'exemple joint à cet article est une petite animation mettant en mouvement un triangle tracé avec les fonctions OpenGL et un dodécaèdre tracé avec une fonction de glut (voir l'écran 2). Le triangle tourne autour de son axe vertical. La dodécaèdre tourne autour des trois axes en mêmes temps. De plus, les deux objets effectuent ensemble un mouvement d'avant en arrière. Le lecteur a toute liberté pour essayer d'autres fonctions de glut pour le rendu de volumes préfabriqués. D'ailleurs, le listing contient quelques lignes de code mises en commentaires spécialement dans cette intention.

Si vous vous souvenez de l'article d'octobre, vous savez que pour initialiser une application glut, il faut définir un certain nombre de fonctions de rappel (*callback*). Pour obtenir une animation il suffit simplement de définir deux fonctions de rappel supplémentaires :

- Une fonction indiquant à glut la conduite à tenir lorsque la fenêtre de l'application est visible ou invisible. Cette conduite étant définie dans une fonction de rappel, on installe la fonction de rappel au moyen de la fonction `glutVisibilityFunc`.
- une fonction indiquant à glut la conduite à tenir quand l'application devient inactive. Cette conduite est définie dans une fonction de rappel.

Voici la fonction de rappel de visibilité de la fenêtre.

```
void visible(int vis)
{
    if (vis == GLUT_VISIBLE)
        glutIdleFunc(idle);
    else
        glutIdleFunc(NULL);
}
```

Cette fonction se contente d'installer ou non la seconde fonction de rappel qui indique à glut quoi faire quand l'application est inactive. On voit que tant que la fenêtre reste visible, glut a quelque chose à faire et rien dans le cas contraire. Cette façon de procéder permet d'être sûr que l'on ne gaspillera pas du temps processeur à calculer une image qui n'apparaîtra de toutes façons pas à l'écran.

Voyons enfin la fonction de rappel «de fin d'activité» :

```
void idle(void)
{
    glutPostRedisplay();
}
```

Dès que notre application entre dans un état d'inactivé, la fonction de rappel `idle` est appelée, celle-ci se contente de poster à l'application un message de rafraîchissement de la fenêtre. Lorsque l'application reçoit ce message émanant d'elle-même elle appelle la fonction *callback* de dessin. Dès que la procédure de dessin est terminée, l'application entre dans un état d'inac-

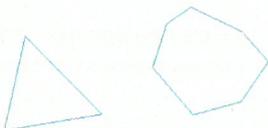
Les polygones convexes

Normalement OpenGL ne sait tracer que les polygones convexes. Nous avons dit qu'OpenGL subdivise en interne les polygones en une série de triangles. Quand le polygone à traiter n'est pas convexe, OpenGL est incapable d'opérer cette subdivision. L'illustration suivante montre des exemples de polygones convexes et non convexes. Si quels que soient deux points pris arbitrairement à l'intérieur du polygone, il est possible de les relier par un segment qui ne sorte pas du polygone, ce dernier est dit convexe. Dans le cas contraire ou dans le cas de polygone «à trou» (exemple de droite dans l'illustration) le polygone est dit non convexe.

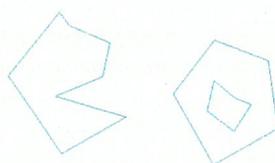
OpenGL offre malgré tout une possibilité de tracer des polygones non convexes moyennant un surcroît de travail de la part du programmeur qui doit préciser pour chaque sommet si celui-ci est un sommet de contour extérieur ou un sommet de contour intérieur, ceci au moyen de la fonction `glEdgeFlag`. Grâce à ces informations supplémentaires, OpenGL se retrouve en mesure de subdiviser

le polygone non convexe en une série de triangles, comme elle le fait pour les polygones convexes. Une épineuse question est de décider s'il est

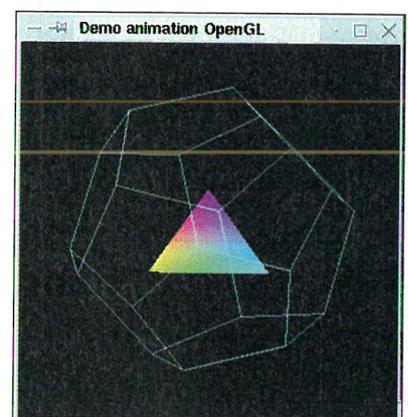
POLYGONES CONVEXES



POLYGONES NON CONVEXES



plus rentable pour le programmeur de fournir à OpenGL les informations permettant le traitement de polygones non convexes ou de subdiviser lui-même ceux-ci en triangles et/ou polygones convexes. Là comme ailleurs, l'expérience et le goût personnels sont les plus utiles pour effectuer le bon choix.



➤ Écran 2: notre programme d'exemple anime un triangle et un dodécaèdre.

tivé ce qui à pour effet de réveiller la fonction `idle`, et ainsi de suite. Nous obtenons donc un effet d'animation.

Retour sur OpenGL, les couleurs

Tous les rendus sont définis au sein de la fonction `draw` du programme d'exemple. Cette fonction ne contient qu'un seul « sandwich OpenGL », ceci pour le tracé du triangle :

```
glBegin(GL_TRIANGLES);
    glColor3f(1.0, 0.0, 1.0);
    glVertex3f(0.0, 0.5, 0.0);
    glColor3f(1.0, 1.0, 0.0);
    glVertex3f(-0.75, -0.5, 0.0);
    glColor3f(0.0, 1.0, 1.0);
    glVertex3f(0.75, -0.5, 0.0);
glEnd();
```

Nous définissons bien 3 vertices en mode `GL_TRIANGLES`, donc un triangle sera dessiné à l'écran. Nous ne reviendrons pas ici sur le système de coordonnées d'OpenGL. En revanche, remarquons que des appels à `glColor3f` s'intercalent entre les définitions de vertices. Nous comprenons ici que lorsqu'un vertex est défini, OpenGL lui attribue la couleur active. En changeant de couleur pour chaque point, on peut obtenir des surfaces multicolores

Pour définir les couleurs, OpenGL présente beaucoup de fonctions qui sont réparties en deux groupes. celles qui reçoivent 3 paramètres et celles qui en reçoivent 4 (voir la [figure 3](#)). Intéressons-nous pour l'instant aux fonctions qui reçoivent 3 paramètres. Ces trois paramètres sont les composantes RGB, c'est-à-dire dans l'ordre *Red*, *Green*, *Blue* (Rouge, Vert et Bleu), les trois couleurs fondamentales de la lumière, à partir desquelles toutes les autres couleurs peuvent être constituées. Chaque paramètre définit le « poids » de chaque composant. Prenons un exemple. Nous décidons d'employer des valeurs flottantes pour définir les composantes de couleurs, c'est donc la fonction `glColor3f` que nous emploierons (Voir *Programmez!* n° 25).

Le maximum d'intensité pour une composante sera représenté par 1.0 et le minimum par 0.0. Ainsi :

```
glColor3f(0.0, 0.0, 0.0);
```

définit la couleur noire

```
glColor3f(0.0, 1.0, 0.0);
```

définit le vert fondamental

```
glColor3f(1.0, 1.0, 1.0);
```

reconstitue la lumière blanche.

The screenshot shows a web browser window with the URL `file:/home/fred/OpenGL/doc/gl/color.html`. The page content is the C specification for OpenGL color functions, listing various `glColor` functions and their parameters.

```
C SPECIFICATION
void glColor3b( GLbyte red,
               GLbyte green,
               GLbyte blue )
void glColor3d( GLdouble red,
               GLdouble green,
               GLdouble blue )
void glColor3f( GLfloat red,
               GLfloat green,
               GLfloat blue )
void glColor3i( GLint red,
               GLint green,
               GLint blue )
void glColor3s( GLshort red,
               GLshort green,
               GLshort blue )
void glColor3ub( GLubyte red,
               GLubyte green,
               GLubyte blue )
void glColor3ui( GLuint red,
               GLuint green,
               GLuint blue )
void glColor3us( GLushort red,
               GLushort green,
               GLushort blue )
void glColor4b( GLbyte red,
               GLbyte green,
               GLbyte blue,
               GLbyte alpha )
void glColor4d( GLdouble red,
               GLdouble green,
               GLdouble blue,
               GLdouble alpha )
void glColor4f( GLfloat red,
               GLfloat green,
               GLfloat blue,
               GLfloat alpha )
void glColor4i( GLint red,
               GLint green,
               GLint blue,
               GLint alpha )
void glColor4s( GLshort red,
               GLshort green,
               GLshort blue,
```

> Figure 3: les fonctions de définition de couleur d'OpenGL.

Les sommets du triangle du programme d'exemple sont chacun d'une couleur différente. c'est la fonction `glShadeModel` qui permet de définir comment OpenGL mélange les couleurs. `glShadeModel` connaît deux modes :

- `GL_SMOOTH`, mode par défaut d'OpenGL crée des dégradés dits dégradés de Gouraud afin d'assurer la transition d'une couleur à l'autre. Chaque vertex peut disposer de sa couleur.
- `GL_FLAT` ou mode plat prend la dernière couleur définie dans un sandwich OpenGL et l'applique à tous les sommets.

Note importante : `glShadeModel` ne DOIT PAS être appelée dans un sandwich OpenGL car une figure ne peut être rendue une partie en Gouraud et une partie en plat.

Les transformations

Abordons maintenant un point délicat de l'infographie 3D : les transformations. Ce sont les opérations mathématiques qui font qu'à partir d'un objet

physique en 3 dimensions on obtient sur l'écran de l'ordinateur une représentation en 2 dimensions donnant une impression de volume et de perspective. Les outils mathématiques, trigonométrie et matrice 4x4 sous-jacents à ces transformations ne sont pas ce qu'il y a de plus simple, loin s'en faut. Heureusement OpenGL est une si bonne bibliothèque qu'une compréhension générale des principes suffit à obtenir des résultats excellents. En effet, OpenGL prend en charge la plus grosse partie du travail, et le programmeur n'a pas besoin de maîtriser le calcul matriciel.

Nous allons donc nous efforcer d'expliquer le plus clairement possible ces principes sans assommer le lecteur avec des considérations mathématiques.

Pour qu'un objet soit finalement représenté sur un écran, il faut que ses caractéristiques physiques soient connues puis transformées mathématiquement. Ces transformations s'enchaînent un peu comme si de l'eau (symbolisant les caractéristiques physiques de l'objet) parcourait une canalisation accidentée pour finalement arriver dans votre baignoire, c'est-à-dire votre ordinateur. Pendant son parcours, l'eau s'est vue raffinée, augmentée de ceci, diminuée de cela, bref, transformée. En fait, cette image de la canalisation est celle employé par les professionnels eux-mêmes qui parlent alors de pipe-line. Notre travail consiste à définir les « accidents-transformations » de la canalisation qui feront que l'eau qui y circule sera modifiée.

Il y a, dans l'ordre, trois types de transformation (voir la figure 4) :

- la transformation de modélisation,
- la transformation de visualisation,
- la transformation de projection.

Ce qui désarçonne souvent le débutant dans ce domaine est que les transformations apparaissent en général dans un programme dans un ordre inverse à celui où elles sont appliquées à l'objet à représenter.

De plus, il arrive souvent que, pour des raisons de performance ou de facilité pour le programmeur, les transformations de modélisation et les transformations de visualisation ne soient pas toujours nettement démarquées les unes des autres dans une applications 3D.

Le programme d'exemple est écrit tout spécialement pour illustrer ces particularités. Nous allons nous concentrer sur le représentation du triangle. Voyons, listing du programme en main, le rôle de chacune des transformations.

Transformation de modélisation

Dans le listing la partie « modélisation » se trouve dans la fonction `draw`. Des « commentaires balises » encadrent le code correspondant. Nous pouvons déjà noter que la fonction `draw` entre en jeu, si l'on considère l'exécution du programme, APRÈS la fonction `reshape` qui s'occupe de transformation de visualisation et de projection. Nous y reviendrons.

La première phase de la modélisation incombe au programmeur, pas à l'ordinateur. Il s'agit de décomposer un volume en sous-volumes élémentaires et aussi le cas échéant de décomposer les surfaces enveloppes de ces sous-volumes en triangles et polygones.

Notre objet est des plus simples puisqu'il s'agit d'un triangle. La première partie de la modélisation pour nous se limite donc à définir/connaître les coordonnées des sommets du triangle.

La deuxième phase de la transformation est, en général, plus complexe. Il s'agit de repositionner par programmation les volumes les uns par rapport aux autres. En effet, il est toujours commode de définir les volumes élémentaires par des coordonnées relatives à l'origine du repère alors que les volumes se positionnent relativement les uns aux autres pour constituer un objet. Une fois les coordonnées définies

par rapport à l'origine du repère, le programmeur doit effectuer un ou des mouvements de translation et rotation pour placer convenablement le sous-volume par rapport aux autres. On emploie pour ceci les fonctions `glRotate` et `glTranslate`. Nous n'expliquerons pas ici ces fonctions qui sont très simples. Le lecteur se reportera à la documentation d'OpenGL et au programme d'exemple.

Par contre, signalons un traquenard qui en a piégé plus d'un : il ne faut pas négliger l'ordre d'appel des fonctions de rotation et de translation. Ainsi une rotation suivie d'une translation donne un résultat tout autre qu'une translation suivi d'une rotation (voir la figure 5).

Pour simuler l'opération de positionnement d'un volume élémentaire dans notre programme d'exemple nous faisons tourner le triangle autour de son axe vertical par un appel à la fonction `glRotate` :

```
glRotatef(angle, 0.0, 1.0, 0.0);
```

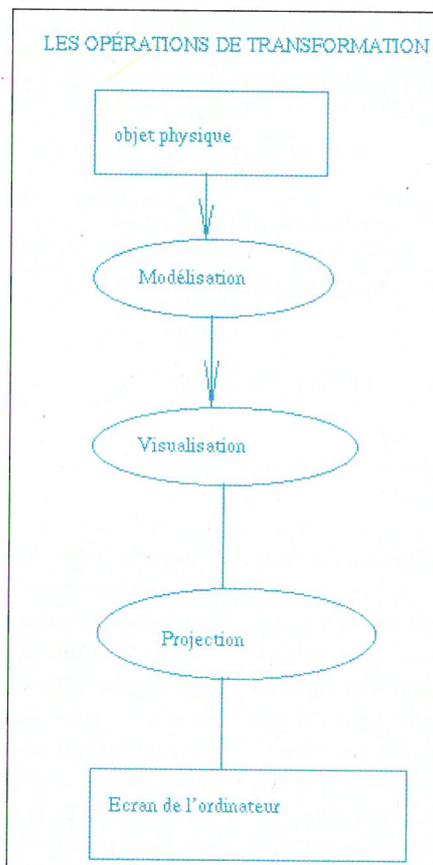
Remarquez bien que cet appel figure dans la fonction `draw` AVANT la définition des points. Il faut comprendre que l'appel à `glRotate` modifie la matrice active de visualisation et que la transformation sera appliquée à tous les vertices définis ensuite et donc injectés dans la matrice *via* le pipe line.

Ici, nous touchons aussi du doigt le problème de la frontière entre modélisation et visualisation, puisque nous venons de parler de matrice de visualisation dans le cadre d'une modélisation.

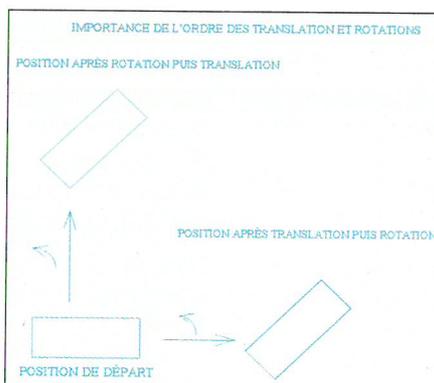
Lorsqu'on souhaite dessiner un objet, dans la vie réelle, on le visualise dans sa tête en même temps. C'est ce que fait le programmeur quand il modélise. Il se représente l'objet, puis dans un second temps, il écrit dans le programme le fruit de son imagination afin que l'application puisse visualiser l'objet pour une autre personne grâce à l'écran de la machine.

Transformation de visualisation

Cette fois, il ne s'agit plus pour le programmeur de visualiser un objet « dans sa tête », mais de définir un point de vue tel que la personne qui regardera l'écran de l'ordinateur aura le sentiment d'être plus ou moins loin de l'objet ou d'être orienté d'une certaine façon par rapport à lui. Il y a deux façons de procéder. La première possibilité est purement OpenGL. L'observateur reste fixe et l'on déplace l'objet ou l'ensemble d'objets au moyen des fonctions `glTranslate` et `glRotate`.



> Figure 4 : les transformations successives subies par les vertices avant d'arriver sur votre écran.



> Figure 5 : un piège classique...

C'est ce que fait le programme d'exemple. Par un appel à `glTranslate` on fait avancer ou reculer l'objet.

À nouveau, il est très important de remarquer que l'appel à `glTranslate` destiné à modifier le point de vue PRÉCÈDE le positionnement des vertices, précédant lui-même leurs définitions. Il faut toujours garder à l'esprit que quand des vertices sont définis par `glVertex` il sont injectés dans la canalisation, et transformés selon les «accidents» survenus auparavant.

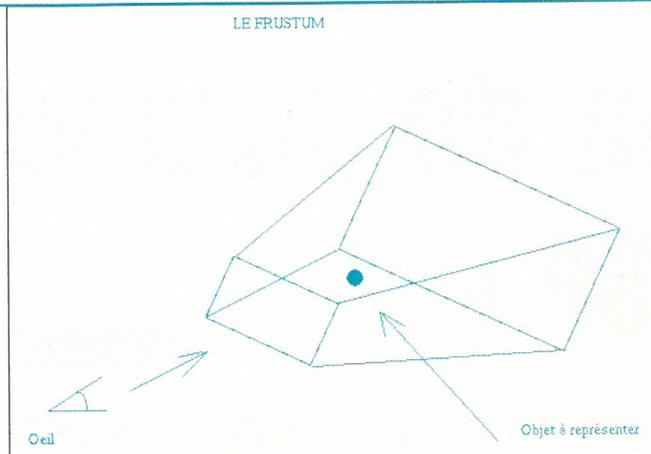
L'autre manière est de changer la position et la direction du regard de l'observateur. OpenGL ne permet pas cela directement. Mais comme nous l'avons déjà dit, OpenGL est toujours accompagnée d'une bibliothèque de fonctions utilitaires de plus haut niveau : GLU, qui propose des fonctions dédiées à la modification du point de vue comme par exemple la fonction `gluLookAt` (regarder à). Nous avons mis en commentaire dans le listing un appel à `gluLookAt` d'un effet strictement identique à celui de `glTranslate`. Les fonctions de la bibliothèque GLU sont d'intéressants raccourcis et allègent la programmation. Le lecteur est invité à consulter la documentation de GLU, normalement jointe à celle d'OpenGL.

Il nous reste à parler d'un dernier point délicat : le *frustum*. Pour qu'une image 3D donne une impression de réalisme il faut obtenir un effet de perspective. Cet effet s'obtient par un calcul matriciel complexe. Une fois encore, OpenGL prend en charge tous ces calculs. Nous n'avons qu'à définir un *frustum* c'est-à-dire un tronc de pyramide dans lequel sera représenté l'objet (voir la figure 6). Les proportions du *frustum* ont une conséquence directe sur l'effet de perspective appliqué à l'objet. Le *frustum* a donc un impact direct sur la qualité de la représentation de l'objet. Là comme ailleurs, choisir un *frustum* est plus une affaire d'expérience qu'autre chose. La fonction qui permet la définition d'un *frustum* s'appelle `glFrustum` et reçoit 4 paramètres.

Ce sont dans l'ordre :

- `left`, qui positionne le bord gauche de la face avant du *frustum* ;
- `right`, qui positionne le bord droit de la face avant du *frustum* ;
- `top`, qui positionne le bord supérieur de la face avant du *frustum* ;
- `zNear`, qui définit l'éloignement par rapport à la position de l'observateur de la face avant du *frustum* ;
- `zFar` qui définit l'éloignement de la face arrière du *frustum*.

Il n'est pas nécessaire que le *frustum* soit un tronc de pyramide symétrique, ni même que son axe soit orienté selon l'axe des z. Le programmeur a toute liberté pour expérimenter et obtenir des effets «intéressants».



► Figure 6 : le frustum est un tronc de pyramide dont les proportions définissent l'effet de perspective appliqué à l'objet représenté.

Pour être effectif un *frustum* doit être validé par un appel à la fonction `glMatrixMode` avec le paramètre `GL_MODELVIEW`. Ceci a pour effet de modifier la matrice de visualisation courante et donc de placer un «accident» sur la canalisation.

Le programme d'exemple définit un *frustum* symétrique orienté sur l'axe des z. À nouveau la définition du *frustum* précède la définition des autres transformations au cours de l'exécution du programme.

Transformation de projection

Nous touchons au but. Il ne nous reste plus qu'à faire en sorte que la représentation 3D de l'objet calculée en interne par OpenGL soit projetée sur l'écran de l'ordinateur c'est-à-dire transformée en une image 2D.

Ceci ce fait dans un *viewport* tel que nous avons appris à en créer précédemment. Une fois le *viewport* défini, il faut en faire un «accident» sur la canalisation comme ceci :

```
glMatrixMode(GL_PROJECTION);
```

La matrice identité

Vous avez sans doute remarqué que le programme d'exemple est parsemé d'appels à la fonction `glLoadIdentity()`.

Lorsque OpenGL calcule des transformations, elle le fait sur une matrice dite matrice courante. De temps en temps, on cherche à cumuler les transformations les unes sur les autres comme c'est le cas dans notre fonction `draw`. Dans d'autres situations on évitera ce cumul, par exemple, entre la définition de la transformation de visualisation et la transformation de modélisation, faute de quoi, à cause du cumul, la transformation de visualisation serait appliquée deux fois ce qui au mieux déformerait l'objet, et au pire, le ferait disparaître de l'écran. Pour éviter ce cumul, on charge dans la matrice courante une matrice neutre, dite «matrice identité», grâce à la fonction `glLoadIdentity`. On peut comprendre ceci comme une remise à zéro ou une réinitialisation avant de nouveaux calculs.

Conclusion

Le lecteur dispose maintenant de toutes les informations de base pour la création et l'animation d'images en 3 dimensions, les autres fonctionnalités d'OpenGL devant simplement en améliorer le réalisme. En attendant de nous retrouver, nous suggérons au lecteur de se faire la main en écrivant, à partir de notre programme d'exemple, le code pour animer une petite pyramide. À bientôt pour de nouvelles aventures dans les profondeurs d'OpenGL. ■

Création de pagelets avec ASP+



Sur votre CD-Rom

Actuellement, le développement d'une application web en ASP n'est pas chose aisée. Graphistes et développeurs doivent apprendre à travailler ensemble : la compatibilité est le maître-mot de cette union.

Par Richard Clark
rc@c2i.fr - <http://www.c2i.fr>

ASP+ permet maintenant de séparer complètement le code du graphisme de la page et ce, de plusieurs façons. La première, que nous avons vue le mois dernier, consiste à placer tout le code dans l'en-tête de la page (entre les deux balises `<head>` `</head>`). Mais vous ne manipulez qu'un seul fichier et l'utilisation d'un utilitaire comme Visual Source Safe devient nécessaire pour une équipe importante.

L'autre façon consiste à placer le code dans un autre fichier. Certaines mauvaises langues me diront qu'on peut toujours le faire avec ASP grâce aux `Include` mais comme vous allez le constater, ASP+ apporte de très nombreux avantages.

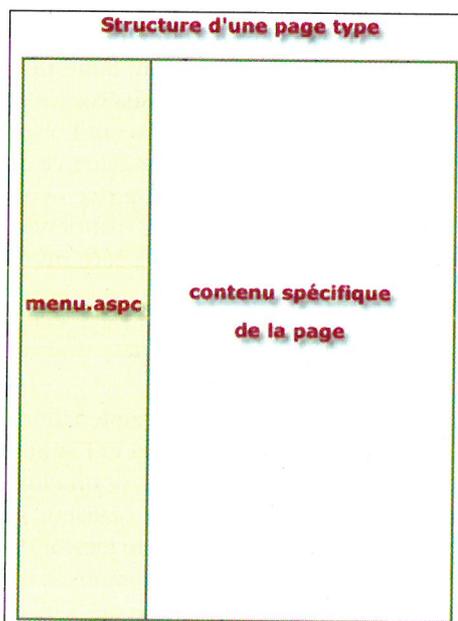
La première solution consiste à utiliser des *pagelets* (petites pages) qui sont une portion de code (HTML ou VBScript) réutilisable comme un `Include` mais en plus puissant.

Une première utilisation basique

Imaginez un site qui affiche dans toutes ses pages sur la gauche un menu.

- En HTML vous pouvez utiliser des *frames* avec les inconvénients que l'on connaît.
- Avec ASP, vous pouvez utiliser `Include`, mais vous risquez de rencontrer des problèmes de conflit de variables, par exemple.
- Avec ASP+, vous utiliserez des *pagelets*.

Comment ça marche? Créez tout d'abord le contenu de votre menu



Écran 1 : l'organisation d'une page...

dans une page nommée `menu.asp` (l'extension `asp`, c comme contrôle, est très importante pour le compilateur). Son contenu ressemblera à cela :

```
Actualités<br>
<a href=»lstnewsASP.asp»>ASP</a><br>
<a href=»lstnews.asp»>Site c2i.fr</a><br>
...
Copyright c2i.fr
```

Les tags `<html>`, `<body>` et autres ne sont pas nécessaires. Maintenant, dans toutes les pages dans lesquelles vous souhaitez voir le menu s'afficher, il suffit de créer son propre `NameSpace` avec un lien vers votre *pagelet*. Pour cela, déclarez votre *pagelet* dans la page comme suit :

```
<@ Register TagPrefix=»c2i» TagName=»Menu» src=»menu.asp» %>
```

Ensuite, à l'endroit où va apparaître votre *pagelet*, saisissez le code suivant :

```
<c2i:Menu runat=»server» />
```

C'est tout.

Le `TagPrefix` permet de créer son propre `NameSpace` (ici `c2i`).

Le `TagName` permet d'identifier un tag dans le `NameSpace`.

Enfin, `src` est le chemin relatif du *pagelet*.

ASP+ fonctionne de la même façon avec ses propres contrôles (`<asp:Label...>`, par exemple).

Le contenu de la page complète possède la structure suivante :

```
<@ Register TagPrefix=»c2i» TagName=»Menu» src=»menu.asp» %>
<html><body>
<table><tr>
<td><c2i:Menu runat=»server» /></td>
<td>...le contenu de votre page</td>
</tr>
</table>
</body></html>
```

Pour l'instant, vous allez me dire que c'est comme un `Include`, mais en plus compliqué ! Regardez plutôt la suite.

Ajout de propriétés

Un autre intérêt d'une *pagelet* est qu'elle peut posséder ses propriétés « personnelles ». Exactement comme les tags HTML, votre *pagelet* peut avoir des attributs.

Imaginez que vous ayez la liste de vos liens préférés à afficher dans plusieurs pages. Avec les *pagelets*, il suffit de créer son propre « contrôle côté serveur » qui possède deux propriétés : `NomDuSite` et `Auteur`. L'utilisation de votre *pagelet* est du style :

```
<c2i:VisuSite NomDuSite=>c2i.fr - Le portail ASP< Auteur=>Richard Clark< runat=>server />
```

Regardez maintenant le code de votre *pagelet*, `visuSite.aspx` :

```
<script language=>VB< runat=>server>>
'On déclare dans un premier temps deux variables internes à notre pagelet
Private sSite As String
Private sAuteur As String
```

Puis, on expose les propriétés à la page qui utilisera la *pagelet*, propriétés en écriture seulement.

```
WriteOnly Public Property NomDuSite As String
Set
sSite = Value
End Set
End Property
WriteOnly Public Property Auteur As String
Set
sAuteur = Value
End Set
End Property
```

Au moment du chargement de notre *pagelet*, on affecte ces propriétés aux deux contrôles `Labels`.

```
Sub Page_Load(Sender As Object, E As EventArgs)
lblSite.Text = sSite
lblAuteur.Text = sAuteur
End Sub
</script>
```

Le contenu qui sera affiché par la *pagelet* apparaît dans l'encadré suivant :

Site : `<asp:label id=>lblSite< runat=>server</>` `
`

Auteur : `<asp:label id=>lblAuteur< runat=>server</>`

Simple non ?

Un autre gros avantage est qu'une même *pagelet* peut être utilisée plusieurs fois dans une même page contrairement aux `Include` (voir le [listing 1](#)).

```
<html><body>
<c2i:VisuSite nomDuSite=>c2i.fr - Le portail ASP< Auteur=>Richard Clark< runat=>server /><p>
<c2i:VisuSite nomDuSite=>Microsoft.com< Auteur=>Bill Gates< runat=>server /><p>
<c2i:VisuSite nomDuSite=>escrot.com< Auteur=>Hacker< runat=>server />
</body></html>
```

Créez vos propres contrôles

Vous pouvez aussi concevoir des *pagelets* plus complexes ce qui vous permet de créer de véritables nouveaux contrôles personnalisés.

Imaginez un formulaire d'inscription comportant de nombreux champs. Parmi ceux-ci, certains doivent être obligatoirement renseignés. Comme vous l'avez vu le mois dernier, vous pouvez utiliser les contrôles ASP+ de validation. Si vous en avez beaucoup, le code de votre page risque d'être chargé et peu lisible (2 contrôles pour un seul champ de saisie). L'idée est donc de créer une *pagelet* qui comportera trois contrôles :

- Un contrôle `Label` qui affichera le titre du champ,
- Un contrôle `Textbox` pour la saisie,
- Un contrôle de validation pour vérifier que la saisie n'est pas vide.

Créez une nouvelle *pagelet* `valid.aspx` :

```
<script language=>VB< runat=>server>>
Private sNomChamps as String

WriteOnly Public Property Champs As String
Set
sNomChamps = Value
End Set
End Property

Sub Page_Load(Sender As Object, E As EventArgs)
lblInfos.Text = sNomChamps
End Sub

</script>
```

le label contenant le titre

```
<asp:Label id=>lblInfos< runat=>Server</> :
'le textbox pour la saisie
<asp:TextBox id=>txtAff< runat=>Server</>
'le contrôle de validation attaché au textbox
<asp:RequiredFieldValidator
id=>checkTextBox<
runat=>server<
controlToValidate=>txtAff<
display=>static<>
Erreur !
</asp:RequiredFieldValidator>
```

Vous encapsulez à l'intérieur de votre *pagelet* trois contrôles à la fois.

Exemple d'utilisation dans un formulaire ne contenant que deux champs obligatoires : Nom et Prénom. On a donc le code très simple suivant :



```
<%@ Register TagPrefix="c2i" TagName="Menu" src="menu.aspx" %>
<html><head>
Sub Page_Load(Sender As Object, E As EventArgs)
v1.Titre = «Liens»
v1.AddURL(«c2i»,»http://www.c2i.fr»)
v1.AddURL(«Microsoft»,»http://www.microsoft.com») v1.AddURL(«Programmez!»,»http://www.programmez.com»)
End Sub
</head><body>
<c2i:Menu id="v1" runat="server" />
</body></html>
```

```
<form runat="server">
<c2i:ValidTextBox id="v1" Champs="Nom" runat="server" />
<c2i:ValidTextBox id="v2" Champs="Prenom" runat="server" />
<input type="submit" value="Soumettre" id="submit1" name="submit1" />
</form>
```

```
Sub btnMonBouton_Click(Sender as Object, e as EventArgs)
....
'votre code quand on clique sur le bouton
....
End Sub
```

Ajout de méthodes

Toujours plus loin, cette fois-ci avec l'ajout de méthodes à votre *pagelet*. Vous allez améliorer le code pour que, désormais, la liste des liens vers les sites puisse se faire dynamiquement avec possibilité de liaison à une base de données.

Cette fois-ci, votre *pagelet* aura l'aspect d'un texte affichant un titre suivi en dessous de la liste de vos liens. Chaque lien possèdera une dénomination (qui sera le texte affiché) ainsi qu'une adresse URL.

Votre *pagelet* aura donc une seule propriété (**Titre**) plus une méthode **AddURL** qui, comme pour un **ListBox** sous Visual Basic, ajoutera un élément dans votre liste :

```
<script language="VB" runat="server">
'Définition de la propriété Titre comme d'habitude
Private sTitre as String
WriteOnly Public Property Titre As String
Set
    sTitre = Value
End Set
End Property
Sub Page_Load(Sender As Object, E As EventArgs)
    If Not Page.IsPostBack then
        lblTitre.Text = sTitre
    end if
End Sub
'Définition de la méthode pour ajouter un lien
Public Sub AddURL(sText as String, sURL as String)
    lblURL.Text += «<br><a href=' & sURL & '>» & sText & «</a>»
End Sub
</script>
'Affichage du titre
<asp:Label id="lblTitre" runat="Server" />
'Affichage des liens
<asp:Label id="lblURL" runat="Server" />
```

Maintenant, le « remplissage » de votre *pagelet*, peut se faire dynamiquement ; voyez par exemple le **listing 2**.

Enfin, comme vous pouvez utiliser des contrôles ASP+ dans vos *pagelets*, vous pouvez gérer les événements de ceux-ci au sein même de la *pagelet*. Si vous incorporez un bouton dans votre *pagelet*, vous pouvez y ajouter le code suivant :

Pour conclure avec les *pagelets* (le nom de *pagelet* n'est d'ailleurs pas définitif, mais pour ma part, je le trouve « sympa »), ce n'est pas juste une nouvelle version des **Incluide**. Elles permettent :

- de séparer le code de la page,
- elles sont paramétrables (propriétés),
- plusieurs instances d'une même *pagelet* peuvent cohabiter dans une même page sans conflit de variables,
- plusieurs *pagelets* écrites en différents langages (VB, C#) peuvent cohabiter dans une même page.

La technique du « Code Behind »

Une autre technique possible avec ASP+ est la création de vos propres composants mais sans avoir besoin d'une quelconque compilation, ni de l'utilisation d'un **regsvr32.exe** pour enregistrer vos classes dans la base de registre.

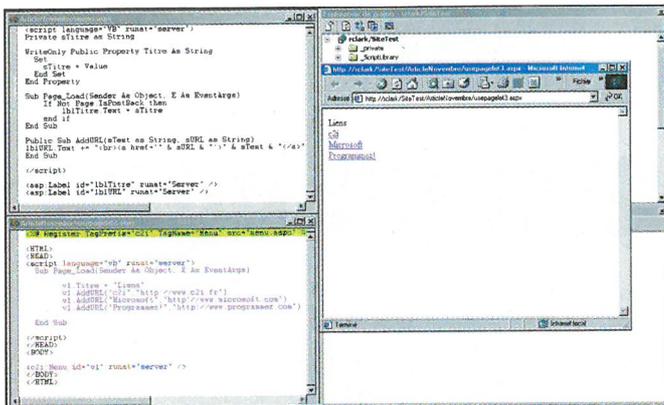
Cette fois-ci, il vous faudra créer des fichiers **.vb** (pour ceux écrits en Visual Basic), **.cs** (en C#) ou **.js** (en Javascript). Comme pour les *pagelets*, l'extension est importante pour que le compilateur ASP+ les reconnaisse.

Ceux-ci contiendront des classes avec des propriétés, méthodes que vous pourrez manipuler dans vos pages.

Prenez l'exemple d'une classe bête et méchante qui sait faire des additions et des soustractions de deux entiers longs (fichier **calcul.vb**) :

```
Public Class cCalcul
Inherits System.Web.UI.Page
    Public Function Add(a as Long, b as Long) as Long
        Add = a + b
    End Function
    Public Function Multi(a as Long, b as Long) as Long
        Multi = a * b
    End Function
End Class
```

J'éviterai de commenter la difficulté de ce code ;-). Le seul point important est de bien préciser que cette classe hérite de la classe **System.Web.UI.Classe** (comme quoi on ne vous a pas menti en vous disant que Visual Basic supportera l'héritage). Cette classe définit toutes les propriétés, méthodes et événements d'une page exécutée sur le serveur.



> Écran 2 : le fichier projet du site IBuySpyVS7.

Pour utiliser un objet de cette classe dans vos pages, il vous faut préciser qu'elle hérite de cette classe. Par exemple :

```
<%@Page Language=»VB» Inherits=»cCalcul» Src=»calcul.vb» %>
<html><head>
<script language=»vb» runat=»server»>
Sub Page_Load(Sender As Object, E As EventArgs)
Dim c as cCalcul
lblInfos.Text = «5 + 5 = « & Cstr(c.Add(5,5)) & «<br>»
lblInfos.Text += «5 * 5 = « & Cstr(c.Multi(5,5))
End Sub
</script>
</head><body>
<asp:Label id=»lblInfos» runat=»Server» />
</body></html>
```

À vous d'imaginer des classes un peu plus sophistiquées qui peuvent contenir vos composants métiers. Vous pouvez inclure votre classe dans votre propre **Namespace**. Pour cela, il suffit, dans votre fichier **.vb** de l'encadrer par la déclaration de votre **Namespace** :

```
Namespace c2i
Public Class cCalcul
...
End Class
End Namespace
```

Pour l'utilisation de votre classe :

```
<%@Page Language=»VB» Inherits=»c2i.cCalcul» Src=»calcul.vb» %>
```

Le code de votre composant est parfaitement lisible et nous verrons dans un prochain article quand nous aborderons les **WebServices** que vous pouvez compiler ce composant sous forme d'une assemblée (fichier **dll**).

```
<%@Page Language=»VB» Inherits=»cBtn» Src=»btn.vb» %>
<html><body>
<form runat=»Server»>
<asp:Button id=»btn1» Text=»Soumettre» runat=»Server» OnClick=»btn1_Click» />
</form>
<asp:Label id=»lblInfos» runat=»Server» />
</body></html>
```

Pour l'instant, vous n'avez qu'un objet externe dont vous pouvez utiliser les méthodes et autres propriétés. Vous constatez qu'il existe toujours dans votre page aspx du code VBScript. Avec la technique du « Code Behind », vous pouvez placer TOUT le code dans ce fichier.

Pour illustrer la chose, considérez une page contenant un bouton et un label qui utilise un fichier **btn.vb** (listing 3). Prenons un exemple simple où le contrôle **lblInfos** affiche « clic » quand l'utilisateur clique sur le bouton. Le fichier **btn.vb** est le suivant :

La première ligne permet d'importer la classe **System** dans votre composant et ainsi savoir à quoi correspond le mot clé

```
Object (entre autres)
Imports System
Public Class cBtn
Inherits System.Web.UI.Button
```

Puis il faut déclarer l'objet **Label** que vous allez manipuler (même nom que dans la page) :

```
Public lblInfos as System.Web.UI.WebControls.Label
```

Enfin, déclarez l'événement et manipulez la propriété **Text** du label :

```
Public Sub btn1_Click(Sender as Object, e as EventArgs)
lblInfos.Text = «Clic»
End Sub
End Class
```

Il n'y a maintenant plus aucune ligne de code dans la page, tout est dans le fichier « **Code Behind** ».

Il y a fort à penser d'ailleurs que cette technique sera la technique privilégiée par Visual Studio.NET. regardez pour vous convaincre le fichier projet Visual Basic du site **IBuySpyVS7** développé par Microsoft.

Finalemnt

ASP+ permet de séparer efficacement le code de l'interface. Certes, cela demande un certain temps d'adaptation mais le jeu en vaut la chandelle pour une meilleure lisibilité et une meilleure maintenance. Si les sociétés (et Microsoft en tête) arrivent à créer des éditeurs ASP+ véritablement WYSIWYG (pas comme Visual Interdev par exemple), les graphistes et les développeurs pourront s'en donner à cœur joie.

Bon développement... ■

3

Comment SDV met les journaux en ligne



Située à côté des Dernières Nouvelles d'Alsace, dont elle est le prestataire de service, la société SDV met les journaux en ligne. Plus d'une dizaine de quotidiens et magazines sont ainsi publiés chaque jour sur Internet par l'entreprise strasbourgeoise, qui est par ailleurs le principal hébergeur institutionnel de l'Est de la France.

Jacques de Schryver
jacques.deschryver@programmez.com

Au début les choses étaient simples. Les DNA et SDV appartenaient au groupe Hersant. Les deux sociétés partageaient des locaux contigus. Et les *Dernières Nouvelles d'Alsace* furent le premier quotidien français en ligne à être mis à jour quotidiennement (septembre 95). Les « Alsaciens du bout du monde », satisfaits de cette formule qui rompait leur isolement, eurent même droit à leur rubrique. Il est vrai que le p-dg de SDV,

Michel Landaret, fut l'un des pionniers du Minitel, dont il inventa et réalisa les premières messageries (Gretel). Aujourd'hui, fort de son expérience, SDV met en ligne de nombreux quotidiens et magazines. Ses clients sont le *Nouvel Observateur*, *Sciences et Avenir*, *Challenges*, *le Figaro*, *l'Est Républicain* et une myriade de quotidiens régionaux. Quels sont les choix informatiques de cette société qui emploie une cinquantaine de personnes ?

Sous Linux depuis 1995

La sécurité y est d'autant plus importante que SDV demeure en outre le premier hébergeur de sites Internet professionnels de la région Alsace, malgré une enclave strasbourgeoise où la Lyonnaise-des-Eaux, ex-Cybercâble, s'est taillé la part du lion. Salim Gasmî, directeur technique de SDV, a opté pour Linux dès son arrivée en 1995. Au

moins quatre raisons ont motivé cette décision. Le choix de Linux semblait évident. Aucune société privée, quelle que soit son importance, ne pourra fournir autant de logiciels que l'*Open Source*, ni avec une telle qualité, ni à des prix aussi bas puisqu'ils sont quasiment gratuits.

On dispose en outre du code source de façon intégrale. On peut donc le modifier, l'adapter. Pour bien comprendre ce point, il suffit de réaliser que dix mille cerveaux valent mieux que cent, qu'ils ont de meilleures idées, et que c'est ainsi que l'*Open Source* évolue, avec des programmeurs bénévoles du monde entier. Qui plus est, Linux est compatible Système V et BSD. Il est aussi stable que les versions payantes d'Unix.

La bibliothèque représente également un argument de poids. Il existe un nombre incroyable de logiciels disponibles dans tous les domaines. Enfin, la gratuité ne peut que séduire.

Salim Gasmi a fait partie des pionniers de Linux en entreprise, avec pour les connaisseurs et les nostalgiques, en juin 95, un Linux Slackware 1.2.13. L'équivalent de l'Apple I pour les précurseurs de la micro-informatique ! Mais Linux n'est pas le seul choix. Aujourd'hui, SDV utilise une vingtaine de configurations PC Intel musclées sous Linux, une dizaine de PC sous Windows NT et une dizaine de serveurs IBM sous OS AIX, donc Unix.

Sécurité : les choix stratégiques

Les PC ont été choisis pour leurs qualités de souplesse et de puissance. Les configurations ont en moyenne de 512 Mo à 1 Go de mémoire vive. Le matériel bénéficie d'un système de sécurité Raid 5 *hot plug hot swap*.

Cela signifie qu'un disque dur qui tombe en panne peut être débranché et remplacé à chaud, donc sans éteindre la machine, grâce aux redondances du Raid 5, et ce, évidemment, sans perte de données.

Chaque PC dispose d'un disque de 30 Go à 500 Go. Les alimentations des machines sont redondantes. Même les



> Salim Gasmi, directeur technique, a choisi Linux dès 1995.

fournir quatre heures de sauvegarde pour l'ensemble des machines. Situé au second sous-sol, il occupe le volume d'un deux-pièces, soit environ cent mètres cube. La salle machine climatisée dispose d'une alimentation en gaz neutre (halon) qui en cas d'incendie étouffe les flammes. Enfin, dernière sécurité, un robot réalise des sauvegardes automatiques régulières dont les cassettes sont rangées dans un coffre anti-feu.

La sécurité des réseaux est assurée selon les mêmes principes. La bande passante est aujourd'hui de cent mégabits par seconde et elle passera à un gigabit avant la fin de 2001. Les routeurs sont des Cisco 2500 à 7200. En cas de panne de l'opérateur, le cas est également prévu. SDV utilise les services de trois d'entre eux simultanément : UUNET, Transpac et Swisscom (BGP4). De cette façon, la sécurité qui constitue un souci majeur de l'entreprise, est assurée. Lorsqu'on héberge une dizaine de quotidiens et magazines, ainsi qu'un certain nombre de grands comptes et de petites industries, la robustesse du service est un argument de choix pour convaincre et garder le client.

« la robustesse du service est un argument de choix »

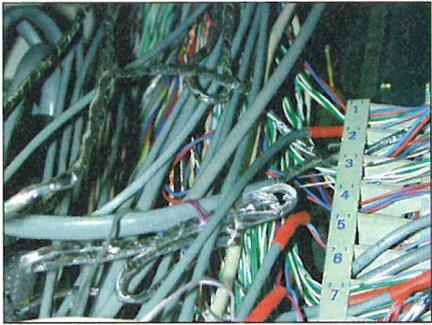
PC en possèdent deux. De même, tous les serveurs et équipements critiques sont doublés. En cas de panne de courant, il existe trois sources d'alimentation électrique différentes, en provenance d'EDS (Électricité de Strasbourg), sans compter l'onduleur gigantesque qui peut

Logiciels et langages

Si le choix de Linux domine chez SDV, d'autres environnements, programmes et logiciels ont droit de cité du fait qu'ils dominent leur catégorie. Il ne s'agit pas de créer une tour de Babel, mais de prendre le meilleur de l'existant, là où il se trouve. Ainsi, un grand soin a été apporté au choix du *firewall*, dont le nom ne sera pas révélé pour des raisons de sécurité. Pour le web, le choix s'est porté sur des serveurs Apache tournant sous Unix et NT4. Le courrier électronique est géré par *sendmail*, le FTP par *FTPD*, les DNS

➔ Gérer les problèmes aléatoires

La mise en place des quotidiens voit le point de rencontre des techniques informatiques et des problèmes d'organisation humaine. Les deux sont correctement maîtrisés, ou nécessitent un temps d'adaptation à la portée de chacun. Mais il existe une catégorie d'obstacles pour lesquels les solutions sont plus aléatoires. Cela concerne, par exemple, les différences d'affichage sur les divers navigateurs. Une page affichée correctement sur l'un ne l'est plus sur l'autre. Et pire, dans un même environnement, l'affichage varie du PC au Mac. Il en résulte un art du compromis intelligent, qui en l'absence de normes liées à la programmation des navigateurs, ne fournit pas de résultats totalement satisfaisants. Cela provoque d'importantes pertes de temps. Pour Christian Neuenschwander, c'est un problème récurrent, même en JavaScript. On peut considérer que cet état de fait est consternant.



➤ Gérer la complexité dans l'enfer des câbles.

par Bind. Enfin, il existe un serveur RealAudio pour le *streaming*. Sous Unix, le développement est réalisé en C, en Shell, avec les nombreux outils internes, avec Python et PHP. Sous NT, les bases de données sont gérées par Cold Fusion (Allaire), SQL et Visual Basic. Oracle et SQL Server jouent également un rôle important. Chaque produit est spécifique. Par

n'existait que ces deux produits. Ils ont été testés et gardés tous les deux. Le choix de VB était davantage contextuel. Il contribue aux petits développements rapides où la performance n'est pas le critère.

Les développeurs

L'équipe de développement, dirigée, pour la partie presse, par Christian Neuenschwander et par Stéphane Adam pour le reste, se compose d'une quinzaine de personnes. Le recrutement concerne des BTS et DUT essentiellement, ainsi que quelques bac + 4



➤ Christian Neuenschwander, responsable de la cellule presse.

ou 5 travaillant dans l'équipe système. Les salaires débutent à 12 500 F brut mensuels sur treize mois pour les BTS et

DUT et à 15 000 bruts mensuels pour les autres. L'évolution de carrière reste ouverte par la suite. D'autant plus que le recrutement pose des problèmes difficilement surmon-

tables, du fait de la pénurie nationale d'informaticiens. En conséquence, les propositions d'emploi sont passées sur le plan national et ne provo-

quent que quatre ou cinq candidatures. Encore chaque candidat a-t-il un vaste choix puisqu'il a de son côté déjà passé entre quatorze et quinze entretiens dans d'autres sociétés!

La formation des nouveaux embauchés leur permet-elle d'être rapidement opérationnels? Pour Christian Neuensch-

wander, l'expérience de l'ASP obtenue à l'université ou dans les écoles d'ingénieurs suffit pour démarrer. Elle permet une transition facile vers le PHP et vers Cold Fusion. Ce dernier est interfacé à SQL Server de Microsoft sous NT. Il est depuis peu disponible en version Linux. Pour le reste, SDV a développé de très nombreuses bibliothèques C, auxquelles les nouveaux venus peuvent collaborer rapidement du fait de leur formation initiale. Des bases de données ISAM fonctionnent ainsi depuis longtemps chez SDV, avec le grand mérite d'avoir été construites sur mesure.

« 1,5 million de visiteurs par mois pour SDV »

exemple, sous NT, Cold Fusion est utilisé pour insérer du code dynamique dans les pages HTML. Même chose sous Unix, avec PHP. Lorsque ces choix ont été opérés, en 1997, il



➤ La salle d'hébergement

Le contact client

Du fait de l'implication presse, qui constitue la vocation première de l'entreprise, SDV a vu d'un œil fort bienveillant l'apparition des nouvelles normes XML. Elles simplifient énormément le travail de transmission des pages, puisque dès lors qu'une société respecte ce format, les données sont déjà compatibles avec le Web. Elles sont affichables directement dans le navigateur. L'AFP, ZD Net pour le *Nouvel Obs* et les *Dernières Nouvelles d'Alsace* nous fournissent déjà leurs pages dans ce format.

Dans de nombreux cas, au contraire, les développeurs de SDV sont contraints de développer des interfaces spécifiques qui peuvent demander

beaucoup de travail. Certains quotidiens ne possèdent pas une structure adaptée au passage vers Internet. Dans une situation précise, tous les articles convergeaient vers une imprimante unique qui centralisait les textes. Il a fallu écrire un logiciel pour récupérer les données à partir du câble RS232C. Il n'existait rien d'autre. Cette solution originale fonctionne encore aujourd'hui, et même s'il s'agit d'une exception, elle donne une bonne idée des problèmes auxquels l'équipe de SDV peut être confrontée.

Une autre difficulté concerne la pagination. En effet, les articles sont parfois envoyés tels quels, charge à SDV de retrouver leur rubrique d'origine. C'est ainsi que, tôt le matin, des salariés lisent la version originale des quotidiens les moins structurés sur le plan de la transmission informatique, afin de rétablir manuellement la cohérence entre les articles et les rubriques. L'une des forces de SDV est

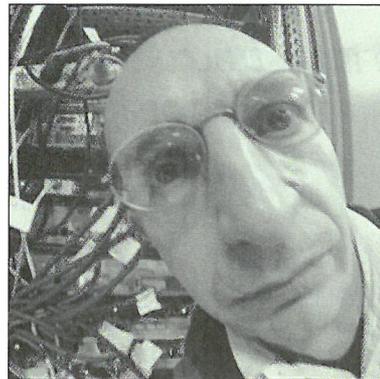
en effet de respecter l'existant plutôt que de vouloir le changer à tout prix. Lorsque les réticences sont trop grandes, lorsque par exemple les journalistes refusent de signaler eux-mêmes systématiquement le nom de la rubrique concernée par leurs articles, alors, on passe à la solution manuelle mentionnée précédemment.

De toute évidence, ce travail aboutit sur la facture du client, qui reste libre d'évoluer comme il le lui est suggéré.

Évoluer avec ses moyens

SDV dispose d'une bande passante importante, et près d'un million et demi de visiteurs accèdent chaque mois à ses pages. Il s'agit bien de visiteurs, car les hits ne sont

pas comptabilisés dans ce calcul. De ce fait, entre bande descendante et bande remontante, qu'il s'agit d'équilibrer, la diversification est venue d'elle-même. Environ six cents entreprises et institutions sont hébergées par SDV, parmi lesquelles l'ENA, Arte, l'Électricité de Strasbourg, l'ASPA (qualité de l'eau), le SDEA (Syndicat des eaux et de l'assainissement) et le Conseil régional. SDV est également le fournisseur Internet de Est Vidéo Communication, qui représente le câble en Alsace (Strasbourg exclu). SDV exploite deux axes de développement dont le potentiel est loin d'avoir été atteint. Il reste beaucoup de magazines et



> Notre reporter perplexe.

quotidiens à mettre en ligne. Et le gros du trafic du commerce électronique sur Internet reste encore à venir. ■

Hors-série

SPÉCIAL PRATIQUE matériels et logiciels !

Sélection des meilleurs dossiers « Pratique » parus dans Compatibles PC Magazine

OFFERT SUR LE CD-ROM :
180 fiches !

> HORS-SÉRIE <
Hors-série n° 36 - Décembre 2000 - Janvier 2001 - 35 F
180 fiches
Sur le CD-ROM

pcmagazine
www.pcmag.fr
le meilleur des articles « pratique » de Compatibles PC Magazine

100% pratique

Devenez un Web créateur
Exploitez les outils pour créer votre site Web (FrontPage 2000, Flash 4, Access...)

Maîtrisez Windows
Tout pour dompter votre environnement (personnaliser vos dossiers, nettoyer votre disque...)

DÉVELOPPEZ VOS PERFORMANCES !

Upgrader facile
Optimisez votre matériel (installez un réseau, une carte mère, un disque dur...)

Animez votre PC
Réalisez vos créations graphiques et musicales (gravez votre plaquette, vos comptes sur CD, vos GIF animés...)

PCmagazine CD-ROM PC
CD-ROM IS 35 DEC 2000 - JANV 2001

FrontPage 2000
de Microsoft • 49 pages, anglaise

PRODUCTION ET CRÉATION PROFESSIONNELLE DE SITES WEB

180 FICHES PRATIQUES
BUREAUTIQUE, INTERNET, GRAPHISME, MATÉRIEL...

Paint Shop Pro 7.0
La toute dernière version du célèbre logiciel de retouche d'images !
• 45 pages, anglaise

SPECIAL MPS
Personnel personnalisé et 2 fonctionnalités

- MPS Station Plus*
- AudioCatalyst 2.1*
- Winamp 2.8S + Pack Fr*
- Media JukeBox 2.2*
- FreeAmp 2.0*
- RealPlayer Basic 8.0

© 2000 Microsoft. Tous droits réservés. Microsoft, FrontPage, Paint Shop Pro et Windows sont des marques de Microsoft Corporation. Windows 2000 et Excel 2000.

EN VENTE DÈS LE 30 NOVEMBRE CHEZ VOTRE MARCHAND DE JOURNAUX



Cocoon

sort de sa chrysalide

Cocoon est une infrastructure de publication XML intégrant la technologie XSP, qui enfilerait bien les beaux habits de standard pour le développement côté serveur.

Gilles Mergoïl, Neoxia

Traditionnellement, les technologies utilisées par les serveurs web intègrent au sein du même document les logiques de présentation (HTML, JavaScript), d'application (Java, VBscript) et d'accès aux données (JDBC, OLE DB).

Certains comme JSP, ASP et PHP incluent des portions de code de programmation dans le document HTML, d'autres comme les *servlets* ou les CGI construisent entièrement le document HTML et le renvoient dans un flux HTTP.

Le modèle proposé par Cocoon est radicalement différent. Il part du principe que le processus de production d'un document web peut se décomposer en trois logiques particulières :

- le document (statique, requête SQL),
- les traitements (filtrage, ajout, ...),
- la présentation selon le format demandé par le client (HTML, PDF, WML).

Chacune de ces trois logiques correspond à des savoir-faire différents. La création du document revient au rédacteur, le traitement est du ressort du programmeur et la présentation est de la responsabilité du *web designer*.

En segmentant ainsi le processus de gestion de l'information d'un site web, Cocoon permet aux différents acteurs de travailler de manière indépendante à la création de composants facilement intégrables et réutilisables.

Les technologies de Cocoon

Cocoon est un projet *Open Source* du groupe Apache XML (xml.apache.org) initié début 1999 par Stefano Mazzochi. Intégralement écrit en Java, il s'appuie sur les principales technologies du W3C que sont XML, XSL, DOM.

Pour fonctionner, Cocoon doit être installé au sein d'un moteur de *servlets* comme, par exemple, Jakarta Tomcat qui est aujourd'hui l'implémentation de référence de la spécification Java Servlet API 2.2 de Sun Microsystems.

Vous l'aurez compris, Cocoon fait une utilisation intensive de la technologie XML et des outils associés que sont XSL, DOM et FOP. Lors de l'installation, le parseur XML Xerces, le transformateur XSLT Xalan et le formateur XSL:FO FOP sont installés dans l'environnement Tomcat ou tout autre moteur de *servlets* implémentant une API 2.x ou supérieure.

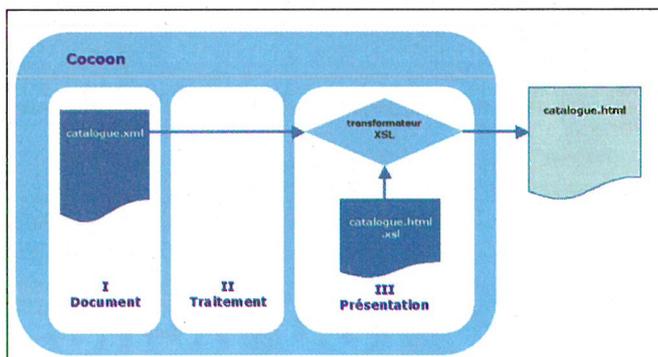
Tout élément de la chaîne de production Cocoon est en fait un document XML, qu'il s'agisse de contenu, des traitements applicatifs ou des transformateurs de format.

Avant de détailler les différentes phases de cette chaîne de production, prenons un exemple simple d'utilisation de Cocoon afin de mettre en évidence les avantages qu'apporte l'infrastructure par rapport aux approches traditionnelles de type JSP ou ASP.

Premier exemple

Considérons un catalogue de produits décrit dans le document `catalogue.xml`. Chaque produit y est décrit par les attributs suivants : référence, nom, prix-vente, photo, descriptif. Il est de la responsabilité du producteur du catalogue des produits qui doit le tenir à jour en utilisant un éditeur XML ou plus vraisemblablement en alimentant la base de données dont il est extrait. Dans le vocabulaire Cocoon, ce document est un producteur et appartient à la **couche I - document**.

Afin de pouvoir afficher le catalogue, le *Web designer* crée le document `catalogue.html.xsl` qui décrit la transformation XSL à appliquer au document `catalogue.xml` pour le transformer en page HTML visualisable sur un navigateur.



> Figure 1 : la réalisation d'un document web met en œuvre trois savoir-faire.

Le document `catalogue.html.xsl` est un formateur et appartient à **couche III - présentation** (voir la **figure 1**).

Une fois que les deux fichiers `catalogue.xml` et `catalogue.html.xsl` sont créés et enregistrés dans le répertoire géré par le *framework* Cocoon, ce dernier prend en charge automatiquement l'ensemble du processus de production du document `catalogue.htm`.

En effet, Cocoon intercepte la demande utilisateur sur le fichier `catalogue.xml`, déclenche la transformation XSLT adéquate et renvoie le fichier HTML généré.

Pour cela, il faut simplement indiquer dans le fichier `catalogue.xml` que celui-ci doit subir la transformation XSL décrite dans `catalogue.html.xsl`. Ceci s'effectue en ajoutant les deux lignes suivantes :

```
<? cocoon-process type="xslt" ?>
<? xmlstylesheet href="catalogue.html.xsl" type="text/xsl" ?>
```

Premières analyses

Bien que cet exemple soit très simple, il met en évidence de nombreux atouts de Cocoon. Aucune ligne de code Java n'a été nécessaire. Le *framework* prend en charge toute la cinématique de transformation ; celle-ci n'a donc plus besoin d'être programmée.

La séparation entre le contenu du document et sa présentation est totale. Ceci permet aux producteurs de contenu et aux *Web designers* de travailler de manières indépendante et parallèle. Il leur suffit de s'échanger auparavant les DTD¹ des documents XML qu'ils traitent.

En corollaire, il est aisé de modifier la provenance des données du catalogue, depuis une base de données par exemple. Cette modification n'affecte pas la couche de présentation.

De la même façon, mettre en place une nouvelle charte graphique pour le site web n'a pas d'impact sur la couche d'accès aux données.

Cocoon apporte également des solutions simples à l'épineuse question des incompatibilités entre navigateurs.

En effet, Cocoon détecte le type et la version du navigateur qui émet la requête HTTP et modifie la chaîne de traitement en conséquence.

Le *Web designer* n'a qu'à fournir une transformation XSL pour chaque navigateur cible et Cocoon utilise la transformation adéquate.

Supposons que la page `catalogue.html` soit différente entre Microsoft Internet Explorer et Netscape Navigator. Il suffit d'écrire les deux transformations XSLT `catalogue.explorer.html.xsl` et `catalogue.netscape.html.xsl` et de modifier le document `catalogue.xml` ainsi :

```
<? cocoon-process type="xslt" ?>
<? xmlstylesheet href="catalogue.html.xsl" type="text/xsl" ?>

<? xmlstylesheet href="catalogue.explorer.html.xsl" type="text/xsl"
media="explorer" ?>

<? xmlstylesheet href="catalogue.netscape.html.xsl" type="text/xsl"
media="netscape" ?>
```

De la même façon, il est tout aussi simple de répondre à une requête émanant d'un téléphone WAP en demandant à Cocoon d'utiliser une transformation XSL vers le protocole WML² !

Enfin, et pour en finir avec les fonctionnalités de la **couche III - Présentation**, Cocoon propose une implémentation de XSL:FO pour le format PDF d'Acrobat, FOP qui permet de générer des documents au format PDF.

Comme nous avons pu le voir sur un exemple simple, l'approche radicalement différente adoptée par le *framework* de publication Cocoon apporte des gains très importants en termes de souplesse et de réutilisation par rapport aux approches traditionnelles.

L'exemple précédent ne mettait en œuvre que des documents statiques. Nous allons maintenant découvrir l'une des fonctionnalités les plus puissantes de Cocoon, la gestion des documents dynamiques et les outils disponibles au sein de la **couche II - Traitement**.

La logique applicative avec les *Extensible Server Pages (XSP)*

Reprenons l'exemple de notre catalogue de produits en le modifiant légèrement.

Il est probable que la base de données catalogue ne stocke pas directement le prix de vente, mais que celui-ci est plutôt le résultat d'un calcul fonction d'un coût de revient et d'un taux de marge.

De plus, lorsque le catalogue est visualisé par le responsable produit, celui-ci voudra certainement connaître la composition du prix et le taux de marge appliqué, alors que ces informations resteront, bien évidemment, invisibles pour le client.

Dorénavant, considérons que l'élément « produit » est décrit par les attributs suivants : référence, nom, coût-revient, marge, prix-vente, photo, descriptif. Où « prix-vente » est le résultat du calcul « coût-revient » x (1 + marge).

Afin, d'une part, de calculer le prix de vente, et d'autre part, d'adapter le document à l'utilisateur (client ou responsable produit), nous allons transformer le document `catalogue.xml` en une page XSP et utiliser un *tag* XSP que nous avons créé pour répondre à la logique applicative souhaitée (voir le **listing 1**).

Le processeur XSP de Cocoon va transformer le document XSP ci-dessus en un nouveau document XML qui sera fonction de l'utilisateur. Dans le cas du client, on obtiendra ce qui figure sur le **listing 2**.

Dans le cas du responsable produit, les deux attributs notés page suivante seront également disponibles :

À propos de l'auteur

Gilles Mergoil est le président de Neoxia, un cabinet de conseil en architecture distribuées.
www.neoxia.com



```
<? cocoon-process type="xsp" ?>
<? cocoon-process type="xslt" ?>
<? xmlstylesheet href="catalogue.html.xsl" type="text/xsl" ?>

<xsp:page language="java"
  xmlns:xsp="http://www.apache.org/1999/XSP/Core"
  xmlns:catalogueXML="http://www.neoxia.com/cocoon/catalogue/XSPtags"
  >

<catalogue>
  ...
  <produit>
    <reference>001</reference>
    <nom>XML editor</nom>

    <catalogueXML:prix-vente cout-revient="100" marge="0.30" />

    <photo>XMLedit.jpg</photo>
    <descriptif></descriptif>
  </produit>
  ...
</catalogue>
</xsp:page>
```



```
<cout-revient>100</cout-revient>
<marge>0.30</marge>
```

Nous pouvons remarquer que le document XML **produit** conserve la directive de transformation XSLT que nous avons étudiée auparavant. Cela signifie que le *framework* Cocoon va de nouveau solliciter le transformateur XSLT pour générer le document HTML final (voir la **figure 2**).

Que s'est-il passé ?

Le processeur XSP a détecté la balise `<catalogueXML:prix-vente>` et a exécuté le code Java qui lui est associé en lui passant la valeur des paramètres coût-revient et marge.

Le code Java, qui fonctionne au sein du moteur de *servlets* a utilisé les informations de la session HTTP pour identifier l'utilisateur et ses droits, et a pu ainsi générer le code XML correspondant à son profil.

Le code Java associé au *tag* XSP `<catalogueXML:prix-vente>` fait partie de la **couche II - Traitement**.

Le programmeur qui est en charge de la logique applicative a mis ce code à disposition de Cocoon dans le répertoire `/cocoon/catalogue/XSPtags`.

Et c'est bien là, la seule référence au sein du document **catalogue.xml** à la logique applicative.

Grâce à l'encapsulation de la logique applicative dans les *tags* XSP, l'indépendance entre les trois couches document, traitement, présentation est conservée.

(1) DTD : *Document Type Definition*

Il s'agit des meta-données décrivant la structure (éléments, attributs, ...) des documents XML.

(2) WML : *Wireless Markup Language*

Protocole basé sur XML adapté aux spécificités de l'affichage sur les téléphones mobiles compatibles WAP (*Wireless Application Protocol*).

(3) LDAP : *Lightweight Directory Access Protocol*

Les technologies sous-jacentes

Sans entrer dans le détail du code Java qui interprète le *tag* XSP ci-dessus, attardons-nous sur certains aspects de ce mécanisme.

Le programmeur responsable de l'implémentation la logique applicative fournit un transformateur XSLT qui interprète le document XML et qui contient du code écrit en Java. En fait, tout langage possédant un interpréteur Java convient.

La spécification XSP comprend un ensemble de balises XML comme `<xsp:logic>` et `<xsp:expr>` qui permettent de mixer le code de programmation et la logique XML.

Cette transformation XSLT est compilée par Cocoon et utilisée lorsqu'un *tag* XSP est rencontré dans le processus de traitement du document.

Grâce à cette technique, les processeurs XSP de la **couche II - Traitement** ont accès à l'ensemble de l'architecture Java. Ainsi, dans l'exemple ci-dessus, les

droits de l'utilisateur ont pu être extraits d'un annuaire d'entreprise *via* une interface LDAP³.

Ou bien encore, il aurait été possible d'accéder aux services applicatifs fournis par des composants EJB dans un serveur d'application.

Enfin, pour terminer sur le sujet des processeurs XSP, il faut savoir que l'on peut intégrer directement des directives XSP et du code Java dans le document XML.

Bien sûr, cette approche rompt l'indépendance entre les couches Document et Traitement, cependant, elle permet d'utiliser la technologie Cocoon avec un existant traditionnel (JSP, *servlets*) et de migrer progressivement vers l'utilisation de *tags* XSP.

Cocoon, situation et perspectives

Cocoon aujourd'hui est disponible en version 1.8 et la version 2 est attendue pour le début de l'année 2001.

Cette nouvelle version introduit des changements majeurs dans l'architecture de Cocoon et ses ambitions sont d'en

```
<? cocoon-process type="xslt" ?>
<? xmlstylesheet href="catalogue.html.xsl" type="text/xsl" ?>

<catalogue>
  ...
  <produit>
    <reference>001</reference>
    <nom>XML editor</nom>

    <prix-vente>130</prix-vente>

    <photo>XMLedit.jpg</photo>
    <descriptif></descriptif>
  </produit>
  ...
</catalogue>
```



faire un *framework* de publication XML de premier plan.

Cocoon 2 systématisé l'utilisation des processeurs XSP et des *tags* XSP. Certains outils comme les *SQL Processors* sont abandonnés au profit de la technologie XSP qui est actuellement soumise au consortium W3C en vue de sa normalisation.

Cocoon 2 propose des avancées importantes dans l'administration des sites web et l'optimisation des performances.

Au sujet de l'administration des sites web, la future version 2 prévoit d'extraire les instructions de traitement (*Processing Instructions*) des documents XML et de les centraliser au niveau du site (*sitemap*). Ceci permettra aux administrateurs web de modifier simplement les stratégies de production des documents.

Dans l'architecture actuelle de Cocoon 1.x, de tels changements nécessitent la modification de chaque document XML. Pour ce qui est de l'optimisation des performances et la capacité à traiter d'important flux XML, Cocoon 2 propose plusieurs évolutions.

Les producteurs de la couche Document et les processeurs des couches Traitement et Présentation seront pré-compilés et participeront à un système de cache évolué.

Les documents XML aujourd'hui traités avec l'API DOM seront traités sous forme de flux grâce à l'API SAX ou JDOM. Cela permettra de restreindre considérablement la quantité de mémoire nécessaire au traitement des documents et donc d'augmenter les performances générales de l'infrastructure.

S'engager avec Cocoon ?

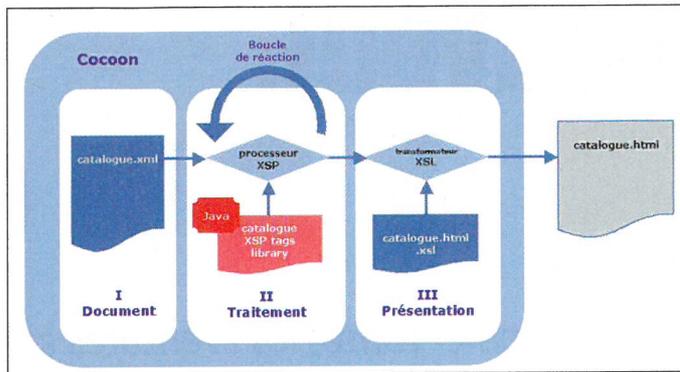
Les quelques exemples évoqués montrent la puissance du *framework* Cocoon dans l'automatisation du processus de production de contenu web.

Faut-il pour cela se jeter à l'eau et oublier les éditeurs HTML et autres ASP ou JSP ? Avant que vous ne franchissiez le pas, faisons quelques remarques.

XML et les outils associés (XSL, XSLT, XSL:FO) sont de nature déclarative et en cela, restent très différents des outils et des langages traditionnels. En effet, XSL s'apparente davantage à Prolog qu'à Java ou PHP ! Cela implique que le coût d'apprentissage de ces technologies est loin d'être négligeable.

Tout changement technologique doit être motivé par une réelle plus-value pour les projets futurs. En d'autres termes, si les technologies que vous utilisez actuellement conviennent à vos besoins et à ceux de vos clients... gardez-les !

Vos équipes et vos clients comprendraient mal les raisons de votre choix si vous livriez votre premier



> Figure 2 : génération du document HTML final via le transformateur XSLT.

projet Cocoon avec un délai de 30% supérieur sans avantage notable en contrepartie.

En revanche, si le nombre de pages de votre site quadruple chaque semestre, que la direction marketing modifie la charte graphique tous les mois et que votre responsable vous a donné six mois pour intégrer les nouveaux clients WAP et télévision interactive... un *framework* XML tel que Cocoon peut vous aider !

La question des performances

Comme souvent, il convient de faire des compromis entre la performance brute d'un système et ses qualités d'évolutivité et d'aptitude à la maintenance.

Accepter une baisse des performances brutes est tout-à-fait envisageable, si par ailleurs, vous réduisez significativement le temps de mise sur le marché de vos nouvelles offres ... tout dépend des contraintes qui vous sont fixées.

Si les contraintes de performance sont trop importantes pour envisager l'utilisation des processeurs XSLT en production, une solution peut être de ne les utiliser que pour la génération du contenu. Les pages HTML restent alors servies de manière traditionnelle par le serveur web, tandis que l'indépendance du contenu et de la présentation sont conservées.

Dans le futur proche, Cocoon 2 promet des améliorations de performance notables, en particulier avec l'abandon de DOM au profit de SAX et la mise en œuvre de systèmes de cache et de pré-compilation.

Conclusion

La technologie XML est aujourd'hui présente dans l'ensemble des offres des principaux fournisseurs de technologies. Il est évident qu'il faudra compter avec XML pour les sites web du futur.

Cocoon est une infrastructure de publication XML très convaincante qui intègre la technologie XSP, prétendant au rôle de standard pour le développement côté serveur.

Le version 1.8 actuelle est fiable et déjà utilisée en production. Assurément, Cocoon 2, annoncé prochainement, sera un digne représentant de l'*Open Source* face aux offres de Microsoft, IBM et consorts. ■

Références

Le site de Cocoon : xml.apache.org
 Le site du W3C et des standards XML : www.w3c.org
 Un site réalisé avec Cocoon : www.eurofootball.com
Java and XML, de Brett McLaughlin chez O'Reilly.

Voici ModLayout.



Ça colle ?

Comme toutes les aventures en Open Source, ModLayout est né d'un besoin. Ce wrapper() s'intercale entre Apache et le cycle de traitement HTML traditionnel. Voici quelques solutions apportées à la composition de pages web.*

par Brian Aker, *Linux Journal*
Traduction : Xavier Leclercq

Vers 1997, je travaillais sur un site nommé Virtual Hospital (<http://www.vh.org>). À l'époque, nous utilisions le serveur Fast Track de Netscape. Je voulais vraiment passer sur Apache, mais je devais au préalable résoudre deux problèmes. D'abord, notre direction devait admettre qu'il serait facile de trouver des gens sachant comment configurer Apache (l'interface graphique du serveur de Netscape semblait faire naître la croyance qu'il était facile à configurer et à maintenir). En second lieu, Fast Track pouvait créer des pieds de page. Le serveur de Netscape supporte les pieds de page en vous permettant d'ajouter arbitrairement du code HTML à la fin de tout document qu'il traite. C'est loin d'être parfait, mais cela suffit aux besoins de base de la plupart des sites.

Le besoin crée la fonction

Le premier obstacle était facile à passer. Vu que nous devions éditer le fichier de configuration du serveur de Netscape de manière régulière, et disposant d'une quantité d'informations minime sur son format fichier, nous avons réalisé que sa maintenance devenait vite une corvée.

Le second problème fut résolu par la création d'un logiciel trappe, un *handler*, personnalisé. La solution était une petite application écrite en C et appelée pour chaque document

HTML. Bien que fonctionnelle, elle présentait un certain nombre d'inconvénients. Ainsi, nous avions maintenant une application qui tournait plusieurs fois par seconde. Cela ne résolvait pas non plus le problème de nos scripts CGI, qui devaient toujours être modifiés pour que le résultat corresponde à la sortie HTML générée par notre *handler*. Cette solution était loin d'être idéale.

Je continuais à avoir des problèmes identiques en travaillant sur divers sites web. Quelqu'un venait me voir, demandait des *copyrights*, des en-têtes ou autres éléments similaires appliqués à l'ensemble des documents présents sur son site ; généralement, il en profitait pour me parler d'une nouvelle technologie de site web, souhaitant utiliser celle-ci pour créer les documents demandés, et ce, sans modifier le contenu de son site. Bien entendu, ce dernier devait pouvoir être facilement remis à jour et modifié. Pour répondre à ces exigences, je créais ModLayout, qui détermine s'il doit traiter un document en examinant le *handler* et le type MIME de celui-ci.

Par exemple, si vous souhaitez traiter des scripts PHP4, vous devrez utiliser la directive `LayoutHandler` avec l'argument `application/x-http-php`. ModLayout traite la majeure partie des types MIME courants sans devoir spécifier les programmes de traitement correspondant dans le fichier de configuration Apache. Ceux-ci peuvent être mis hors service à l'aide de « `LayoutDefaultHandlers-Off` », et peuvent être rajoutés plus tard si vous le désirez.

Des sites auront souvent des documents spécifiques qui ne doivent pas être traités. Pour atteindre cela, vous devez utiliser la directive « `LayoutIgnoreURI` ». Il existe aussi des directives d'omission pour n'enlever que les en-têtes, les pieds de page ou les en-têtes http.

(*) ModLayout est ce que l'on nomme un « wrapper », c'est-à-dire une « colle logicielle » qui s'intercale entre Apache et le cycle de traitement HTML traditionnel. Il existe d'autres wrappers sous Linux, comme le TCP-WRAPPER qui s'intercale entre le flux TCP/IP et le noyau pour filtrer les services réseaux.

ModLayout adjoint trois phases supplémentaires au processus Apache. Il fournit une phase d'en-tête, une phase de pied de page et une phase pour créer des en-têtes http. Grâce à ModLayout, vous pouvez si vous souhaitez créer dynamiquement des en-têtes http pour tous vos documents. Sur une page HTML, les en-têtes et les pieds de page se présentent avant et après un document original.

Il existe trois types d'en-têtes et de pieds de page. Vous pouvez ajouter un simple extrait de texte :

```
LayoutHeader "Sample header"
LayoutFooter "Sample Footer"
```

vous pouvez aussi insérer des fichiers entiers en utilisant :

```
LayoutHeader /usr/local/apache/htdocs/header.html
LayoutFooter /usr/local/apache/htdocs/footer.html
```

ou au moyen d'URI :

```
LayoutHeader /header.pl
LayoutFooter /footer.php
```

Les exemples ci-dessus fonctionneront si vous avez des besoins simples pour vos en-têtes et vos pieds de page. Les deux premiers exemples fonctionnent également parce que

leur contenu est mis en antémémoire dans le processus Apache. Par contre, si vous avez besoin d'un élément plus dynamique, vous pouvez recourir à l'exemple URI. ModLayout ne tient pas compte du type d'URL utilisé. Cela peut aller de simples CGI et SSI à des applications complexes écrites en Perl, PHP ou Java. Les modules personnalisés écrits en C à l'aide de l'API d'Apache peuvent également être utilisés. Si vous désirez utiliser uniquement du code HTML sans le mettre en antémémoire, vous pouvez simplement spécifier son URI et pas son chemin d'accès complet. ModLayout fournit aussi des variables d'environnement supplémentaires pour les URI appelés comme des en-têtes et des pieds de page. Ces derniers offrent des informations sur l'état du document d'origine. Un des principaux avantages de ModLayout est qu'il fonctionne avec toute forme de contenu et que son langage est neutre.

Une utilisation commune de ModLayout consiste à attacher un avis de *copyright* à l'ensemble des documents présents sur un site. Pour cela, vous pouvez recourir à ce qui suit :

```
<VirtualHost www.example.com>
  <...insert normal content...>
  LayoutFooter " <P>Copyright Example.com 1995-200 "
  LayoutHandler "application/x-httpd-php"
</VirtualHost>
```

BIEN DÉMARRER AVEC LINUX !

Linux+ vous offre un « kit d'initiation complet » :

- ✓ Installation ✓ Configuration
- ✓ Trucs et astuces ✓ Bureau KDE
- ✓ Fonctions Multimedia ✓ Connexion Internet
- ✓ Installation des applications.

Le magazine est accompagné de 2 CD-Rom avec les outils indispensables :

- **CD-Rom n°1**
COREL LINUX dernière version
- **CD-Rom n°2**
STAR OFFICE 5.2, COREL PHOTO PAINT, WORDPERFECT 8, KOFFICE...

**UN RENDEZ-VOUS
À NE PAS
MANQUER !**

HORS-SERIE
Linux+ Exploiter le meilleur de Linux
117 Novembre/Décembre 2000 - 80 F

STARTER KIT LINUX

1 LE SYSTEME
Corel Linux OS
Second Edition

2 LA SUITE BUREAUTIQUE ET LES OUTILS
StarOffice 5.2

3 LE MAGAZINE
52 pages pour bien démarrer

ETAPE PAR ETAPE
 ■ Installation ■ Configuration
 ■ Trucs et astuces ■ Bureau KDE
 ■ Multimedia ■ Internet
 ■ Installer les applications

CD N°1
LE SYSTEME LINUX
Corel Linux OS
Second Edition
Nouvelle version
Installation simple
Bureau KDE amélioré
Gestion de port USB
Support des cartes graphiques racantes
Netscape Communicator intégré

CD N°2
LES APPLICATIONS
StarOffice 5.2
Suite Bureautique complète !
Traitement de texte ■ Tableur
Agenda ■ Base de données
Présentation

Corel Photo-Paint 8
Création d'images bitmap et retouche
COMPLÉT 30 JOURS EN LIGNE

OpenOffice.org pour Linux
KOffice ■ GeniusCart Pro ■ Mozilla

EN VENTE ACTUELLEMENT CHEZ VOTRE MARCHAND DE JOURNAUX

```

<VirtualHost 205.153.154.37:80>
#####
##### www.tangent.org
#####
ServerName www.tangent.org
RandomQuote On
RandomQuoteFile /usr/local/apache/conf/quotes
DocumentRoot /usr/local/apache/htdocs-Tangent
UserDir public_html
CustomLog /usr/local/apache/logs/www_tangent_org.access combined
ErrorLog /usr/local/apache/logs/www_tangent_org.error
<Directory /usr/local/apache/htdocs-Tangent/>
LayoutHeader "/usr/local/apache/htdocs-Tangent/header.html"
LayoutFooter "/footer.pl"
LayoutHandler httpd/unix-directory
</Directory>
<Directory /usr/local/apache/htdocs-Tangent/mod_layout/>
LayoutHeader
"/usr/local/apache/htdocs-Tangent/mod_layout/header.html"
</Directory>
<Directory /usr/local/apache/htdocs-Tangent/mod_random/>
LayoutHeader "/usr/local/apache/htdocs-Tangent/mod_random/header.html"
</Directory>
</VirtualHost>

```

L'exemple ci-dessus insérerait le texte au bas de chaque document sur l'hôte virtuel www.example.com pour l'ensemble des programmes de traitement par défaut, y compris les pages HTML et les scripts CGI, ainsi que sur tout script PHP4 présent. Si vous souhaitez ajouter la date à laquelle le document d'origine a été modifié pour la dernière fois, vous pouvez insérer la séquence suivante au sein d'un document PHP, et le sauver en tant que « last-modified.php ».

```

< ?php
    echo(" Copyright 2000 FooBar INC. <BR> Last Modified");
    echo ($LAYOUT_LAST_MODIFIED);
?>

```

et modifier la directive `LayoutFooter` en `/last_modified.php`.

Plus compliqué

Passons à un problème plus complexe. Imaginons que vous vouliez créer sur un site bien précis un « *look and feel* » personnalisé qui a été imposé par le serveur web, ce qui suit pourrait vous servir d'exemple :

```

<VirtualHost www.example.com>
<...insert normal content...>
LayoutHeader "/header.php"
LayoutFooter "/footer.pl"
LayoutIgnoreHeaderURI "/index.html"
LayoutHandler "application/x-httpd-php"
</VirtualHost>

```

Dans l'exemple ci-dessus, le site présenterait un en-tête personnalisé créé par un script PHP et un script Perl fonctionnant pour le pied de page. Tous les *handlers* par défaut sont traités ainsi que tout script PHP présent sur le site. Remarquez que le document `index.html` ne sera pas traité. En effet, la plupart

des sites présentent une première page personnalisée mais veulent avoir un aspect par défaut sur les pages internes du site. Vous pouvez aussi utiliser une espèce de `mod_random` pour créer votre propre système d'ajout de bannières.

Lorsque vous utilisez `mod_random`, souvenez-vous lors de la création d'en-têtes de prendre garde à ne pas créer du code HTML invalide (heureusement la plupart des navigateurs le traiteront). Néanmoins, si vous imposez une présentation spéciale sur des pages créées par des utilisateurs extérieurs, ils risquent de se retrouver avec un résultat désordonné. Gardez en mémoire que l'insertion de tout code HTML avant une page munies de *frames* interrompra ces dernières. Pour l'instant, la solution employée par la majorité des gens est l'utilisation de pieds de page et/ou la création de boîtes JavaScript flottantes. Dans une version future, vous pourrez mettre hors service en-têtes et pieds de page à l'aide de directives reçues d'un URL appelé à partir d'un en-tête ou d'un pied de page.

Pour les sites qui ont des besoins plus complexes, consultez le [listing 1](#), plus sophistiqué.

L'exemple fourni dans ce listing est utilisé couramment pour <http://www.tangent.org/>. Cela permet pour le site web une mise en traitement de documents qui n'affectent que le contenu du répertoire principal racine mais délaisse la page d'accueil des utilisateurs. Bien que le site tout entier utilise le même pied de page, l'en-tête a été modifié pour les différentes sections. ModRandom est utilisé pour insérer des séquences « *random quotes* » au sein des pieds de page (heureusement, ceux-ci proviennent du dessin animé « *The Simpsons* » et n'ont rien d'aussi ennuyeux qu'une bannière publicitaire).

Le contrôle de ModLayout avec les *tags* `Directory`, `VirtualHost` et `Location` vous permet de résoudre des logiques assez compliquées. Vous pouvez également contrôler ce qui crée les en-têtes http à travers `LayoutHTTPOverride` et en créant vos propres en-têtes http dynamiques à l'aide de `LayoutHTTPHeader`. Pour l'instant, il y a environ 24 directives totales et ModLayout continue à se développer de manière assez active, (le développeur diffuse son produit environ une fois par mois et des diffusions complètes sont réalisées tous les deux mois.) Des suggestions de caractéristiques à ajouter sont encouragées et il existe un forum afin de pouvoir discuter des caractéristiques, des HOWTO et des bogues. Pour plus d'informations, consultez la page d'accueil de Modlayout, http://tangent.org/mod_layout/.

Bien que ModLayout ait été créé pour résoudre un de mes problèmes, il est de nos jours tout autant le résultat de personnes qui m'écrivent et demandent de nouvelles caractéristiques. Sa direction future dépendra pour beaucoup de ce que les gens me demanderont et de ce dont ils auront besoin. ■

À propos de l'auteur :

Brian Aker consacre une partie de son temps à un BBS, le célèbre <http://www.slashdot.org>, où il s'occupe des bases de données et d'Apache. Il est également l'auteur de `mod_random` et d'autres modules Apache.



Attention,

Pas sécurisé, Windows NT ? Pas si sûr. S'il est certain qu'un système « out of the box » fait penser à un certain fromage à trous, il suffit d'un peu de pratique et d'une bonne dose de méthode pour obtenir un système déjà beaucoup plus solide et adapté à nos besoins de sécurité !

Patrick Chambet
patrick.chambet@edelweb.fr

Windows NT 4.0 comporte de nombreuses fonctionnalités de sécurisation. Cependant, lors d'une installation par défaut du système, la configuration de ces fonctionnalités est laissée trop lâche. Avec Windows 2000, Microsoft a tenté de remédier à ce problème, et propose des configurations par défaut plus robustes, mais qui sont encore trop orientées vers la facilité d'utilisation plutôt que vers la sécurité du système et la protection des données. L'objectif de cette série de deux articles est de décrire les évolutions de Windows 2000 par rapport à Windows NT 4.0 en matière de sécurité, et de présenter des recommandations de sécurisation concernant les deux systèmes en vue d'obtenir un serveur correctement sécurisé.

La démarche de sécurisation

Dans le cas d'une entreprise, une démarche de sécurisation commence par la rédaction d'une politique de sécurité. Les étapes nécessaires pour cela sont entre autres les suivantes :

- bilan de la stratégie de l'entreprise, de son organisation et de ses processus ;
- analyse des risques (évaluation des menaces et des vulnérabilités) ;
- définition d'une architecture de sécurité, rédaction d'un schéma directeur.

Ensuite seulement vient l'implémentation de la sécurité, suivie au besoin

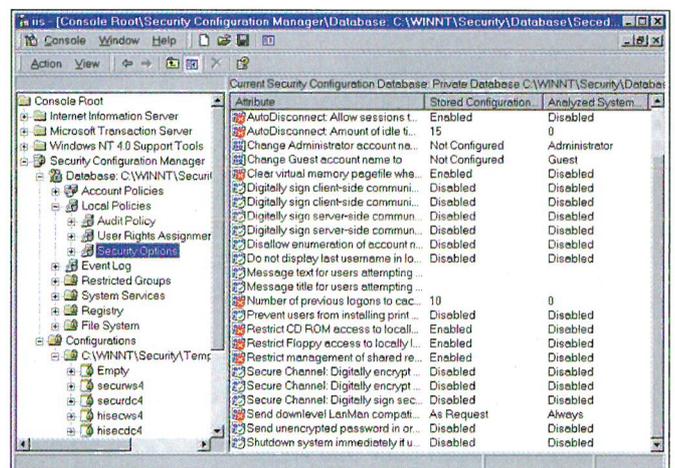
par un audit de configuration puis par des tests d'intrusion. Dans le cas d'un particulier, celui-ci devra définir ses besoins (plusieurs utilisateurs partagent l'ordinateur,...) et ses risques (accès Internet permanent, ...).

Les niveaux de sécurité

Windows NT 4.0 et Windows 2000 permettent d'établir différents niveaux de sécurité, allant d'un niveau de sécurité faible à une sécurité élevée. Pourquoi ne pas choisir systématiquement d'avoir une sécurité maximale ? Une raison est que plus un système est protégé, plus les tâches des utilisateurs deviennent complexes à accomplir. De plus, la mise en place et la gestion des protections constituent une charge supplémentaire pour les administrateurs.

D'autre part, si la sécurité est trop stricte, les utilisateurs peuvent être tentés de la contourner pour faciliter leur travail (mot de passe écrit sur un Post-It sous le clavier, pas de fermetures de sessions, etc.).

Il est donc nécessaire d'évaluer votre besoin de sécurité afin de trouver le juste équilibre entre niveau de sécurité et facilité de travail pour les utilisateurs avant de mettre en œuvre



> Écran 1 : l'interface graphique de gestion centralisée du système est apparue avec l'option Pack de Windows NT4.0

zone protégée !

une configuration de sécurité. De plus, la mise en œuvre de paramétrages de sécurité ont un impact direct sur la configuration du système. En particulier, certaines applications exigeront peut-être des paramètres plus souples afin de fonctionner correctement. Il faut donc évaluer avec soin chacune des recommandations préconisées dans cet article en la mettant dans le contexte de votre système afin de s'assurer que celui-ci fonctionnera toujours après l'application de la recommandation. Le résultat de cette phase d'analyse et de tests pourra être la réalisation d'un *master* permettant d'obtenir un système configuré pour assurer une sécurité définie comme minimale.

Sécurité physique

La pièce dans laquelle se trouve l'ordinateur, surtout lorsqu'il s'agit d'un serveur, devra être fermée à clé lorsque personne n'est là pour le surveiller afin d'éviter le vol. L'idéal serait que cette pièce se trouve dans un bâtiment dont l'accès est autorisé aux seules personnes habilitées.

Dans le cas d'un ordinateur portable, utilisez un câble anti-vol pour l'assujettir à un point solide.

Environnement

Le déni de service le plus évident est l'interruption du courant électrique. Utilisez une protection contre les surtensions et/ou un onduleur pour préserver l'ordinateur et éviter toute perte de données ou altération des partitions au cours d'un arrêt brutal.

Faites également attention aux risques de type incendie ou dégâts des eaux en respectant les normes de locaux en vigueur.

Politique de sauvegardes

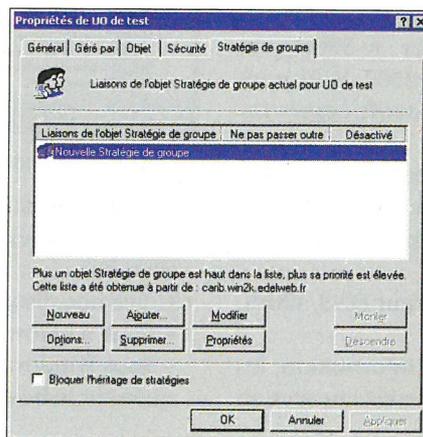
Des sauvegardes régulières doivent être effectuées, afin de protéger vos données des pannes matérielles, des erreurs de manipulation, des virus et autres dommages délictueux.

Un ou plusieurs jeux de sauvegardes devront être conservés dans des lieux géographiques différents. Ces sauvegardes elles-mêmes devront être sécurisées (attention au transport).

Contrôle d'accès à l'ordinateur

Il est impossible de sécuriser totalement un ordinateur si on peut y accéder physiquement. Toutefois, il est conseillé de prendre les mesures suivantes :

- Si l'ordinateur est une station de travail, configurez un mot de passe au *boot* dans le BIOS, et protégez également la configuration de celui-ci par un mot de passe d'administration. Dans le cas d'un serveur, si vous voulez qu'il redémarre automatiquement en cas de coupure de courant par exemple, il ne faut pas mettre de mot de passe de *boot*.



> Écran 2 : les propriétés définies pour une stratégie de groupe.

tegez également la configuration de celui-ci par un mot de passe d'administration. Dans le cas d'un serveur, si vous voulez qu'il redémarre automatiquement en cas de coupure de courant par exemple, il ne faut pas mettre de mot de passe de *boot*.

- Évitez les *multi-boot*, qui permettent de démarrer l'ordinateur avec un autre système d'exploitation (Linux, ...).
- Désactivez l'amorçage par disquette dans la configuration du BIOS. Si possible, retirez le lecteur de disquettes, afin d'empêcher le démarrage sur un système d'exploitation permettant d'utiliser des utilitaires comme NTF-

SDOS pour accéder aux partitions NTFS.

- Si l'ordinateur possède une serrure physique, verrouillez-la et conservez la clé dans un endroit sûr. Toutefois, si la clé est perdue ou inaccessible, il se peut qu'un utilisateur autorisé soit dans l'incapacité de travailler sur l'ordinateur. Attention donc aux auto-dénis de service induits par des procédures mal définies.
- Toutes les partitions du disque dur doivent être au format NTFS.



Les nouveautés de W2K

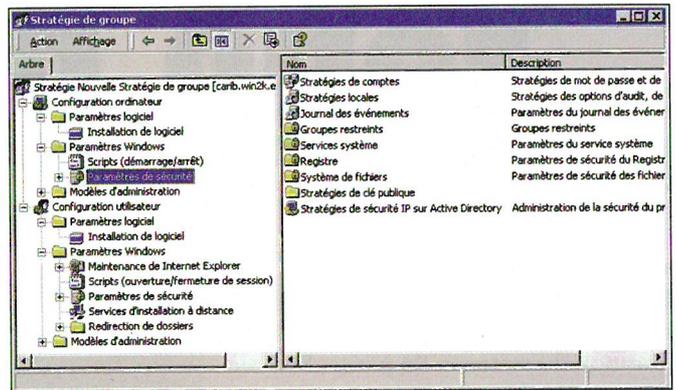
Les nouvelles fonctionnalités concernant la sécurité, par rapport à Windows NT 4.0, sont les suivantes :

- Sécurité dans *Active Directory* ;
- Stratégies de Groupes ;
- Modèles de sécurité ;
- Autres possibilités (suppression de NetBIOS, ...)
- Authentification par Kerberos V.5 ;
- Permissions d'accès aux fichiers plus fines ;
- Paramètres de la base de registre plus sécurisés ;
- *Security Configuration Tool Set* et *Secedit* ;
- *Encrypting File System* (EFS) ;
- *IP Security* (IPSec) ;
- Protection des fichiers système (WFP) ;
- Support des PKI intégré ;
- Support de *Smart Cards* intégré ;

Installation de Windows NT 4.0 et W2K

Les étapes à suivre pour l'installation de Windows NT 4.0 ou Windows 2000 à partir des CD-ROM originaux sont les suivantes :

- Pour un serveur ne devant pas faire partie d'un domaine, choisir une installation en serveur autonome dans un groupe de travail.
- Installez ensuite les derniers *Service Packs* : le SP6a pour Windows NT 4.0 et le SP1 pour Windows 2000. Vous pouvez les télécharger à : <http://www.microsoft.com/windows2000/downloads>
- Installez ensuite les derniers *hotfixes*. Attention, ceux-ci nécessitent une installation dans un ordre chronologique précis. Vous pouvez les télécharger à : <http://windowsupdate.microsoft.com>
- Puis installez les services nécessaires : serveur DNS, WINS, DHCP, RAS, PPTP, etc.
- Arrêtez les services non utilisés. À titre d'information, les services indispensables pour faire fonctionner une station de travail Windows 2000 sont les suivants :
 - *Event Log*,
 - *NT LM Security Support Provider*,
 - *Remote Procedure Call (RPC)*,
 - *Security Accounts Manager*,
 - *Workstation*.
- Notez que même le service Server est arrêté. Si vous n'avez pas de partage réseau, ce n'est pas un problème. Par contre, l'explorateur d'ordinateurs sera arrêté également, et vous ne pourrez plus lister les ordinateurs de votre domaine.
- Désinstallez les protocoles réseaux non utilisés (IPX/SPX, NetBEUI).
- Désactivez les services réseaux non utilisés sur certaines cartes réseaux (typiquement NetBIOS si on n'a pas à accéder à des partages réseaux sur un LAN).
- Installez vos applications.
- Ré-appliquez le dernier *Service Pack* et les *Hotfixes*. Ce dernier point est obligatoire sur NT 4.0. Sur Windows 2000, le mécanisme de mise à jour utilisant Windows Installer et WFP (*Windows File Protection*) fait qu'il n'est plus nécessaire de repasser les *Service Packs*.

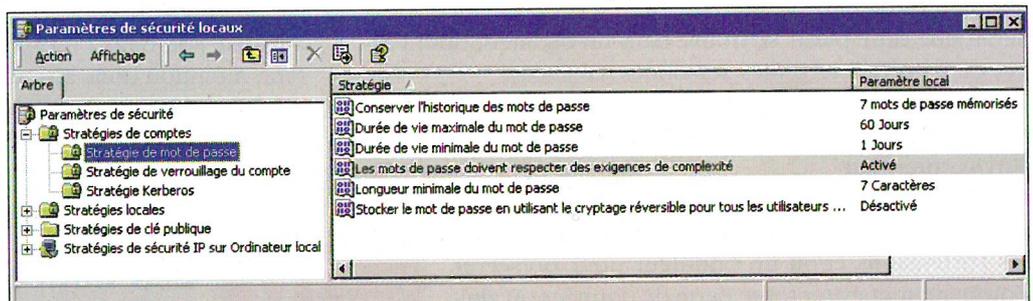


> Écran 3 : il est possible de créer une stratégie pour chacun des conteneurs.

- Testez votre serveur et vérifiez qu'il peut communiquer à travers le réseau et que les applications et services fonctionnent.
- Créez une disquette de réparation d'urgence :
 - Sous Windows NT 4.0, tapez `rdisk /s`
 - Sous Windows 2000, utilisez l'utilitaire de *backup* et cliquez sur le bouton « **Disquette de réparation d'urgence** ».

Paramétrage du système

L'application des paramètres de sécurité qui vont suivre dans cet article et le suivi des serveurs une fois configurés est laborieux car actuellement, sous Windows NT 4.0, il n'existe pas d'outil centralisé pour gérer la sécurité d'une station ou



> Écran 4 : Le paramétrage des mots de passe.

1- Stratégie de mots de passe :

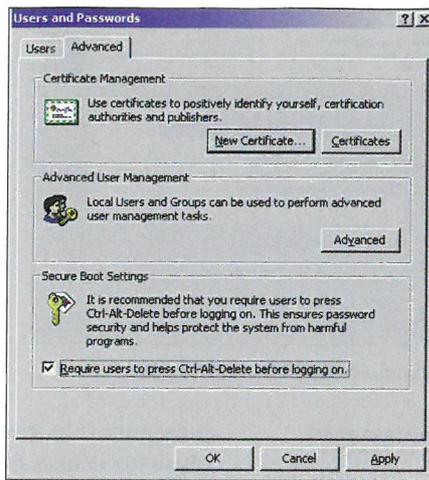
Durée de vie maximale du mot de passe	60 à 90 jours.
Durée de vie minimale du mot de passe	Bien que ce paramètre semble peu important, s'il n'est pas défini, des utilisateurs pourront contourner la condition de non-réutilisation de leurs anciens mots de passe en changeant immédiatement leur mot de passe plusieurs fois de suite de façon à retomber sur leur mot de passe initial. Paramétrer à un jour minimum.
Longueur minimale du mot de passe	7 (ni 6, ni 8).
Conserv. l'historique des mots de passe	6 ou 7.
Les mots de passe doivent respecter des exigences de complexité (Windows 2000 seulement)	Activé.
Stocker le mot de passe en utilisant le cryptage réversible pour tous les utilisateurs du domaine (Windows 2000 seulement)	Désactivé.

Attention, zone protégée !

d'un serveur : l'administrateur jongle en général avec le gestionnaire des utilisateurs, l'éditeur de stratégies, l'explorateur NT, l'éditeur de registres, le journal des événements et le panneau de configuration des services, auxquels il rajoute en général ses propres outils.

Windows 2000, par contre, est livré en standard avec un outil permettant de gérer depuis un seul et même endroit tous les paramètres liés à la sécurité du système local : l'outil d'administration nommé « Stratégie de sécurité locale » (et « Stratégie de sécurité du domaine » dans le cas d'un contrôleur de domaine). Il est même possible d'ajouter dans cet outil ses propres paramètres de sécurité (les options de sécurité correspondant à des clés de la base de registre sont situées dans

la clé `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SeCEdit\Reg Values`). Windows 2000 dispose également d'un outil permettant de définir des profils de sécurité, d'analyser un système par rapport à ces profils et d'appliquer ceux-ci au système : il s'agit de l'outil « Configuration et analyse de la sécurité ».



> Écran 4 : le panneau de configuration des mots de passe.

Les utilisateurs de NT 4.0 peuvent utiliser un outil de ce type : le « *Security Configuration Tool Set* », livré sur le CD-ROM du SP4 pour NT 4.0, dans le répertoire `MSSCV386` (bien qu'il ne soit pas installé par défaut en même temps que le SP4), ou téléchargeable depuis le site de Microsoft :

<ftp://ftp.microsoft.com/bussys/winnt/winnt-public/tools/scm>

Le *Security Configuration Tool Set* est un « composant enfichable » qui s'installe dans la MMC (*Microsoft Management Console*), l'interface graphique de gestion centralisée du système, introduite sur NT 4.0 avec l'Option Pack et qui est le cœur de la gestion de Windows 2000 (voir l'écran 1).

Cet outil permet de :

- définir ses modèles de sécurité, en plus des 10 modèles fournis en standard ;
- analyser et contrôler la configuration courante du système et la comparer à un modèle de sécurité ;
- appliquer au système les paramètres de sécurité définis dans un modèle.

Choisissez la formule gagnante

Compatibles PC Magazine évolue pour offrir aux utilisateurs exigeants que vous êtes l'information que vous attendez.

- ✓ PLUS D'INFORMATIONS
- ✓ PLUS DE TESTS
- ✓ PLUS D'INTERNET
- ✓ PLUS D'UPGRADE
- ✓ PLUS DE PRATIQUE

Avec chaque numéro un CD-ROM exclusif avec des programmes complets, des utilitaires, des sharewares, des démos jouables... pour aller plus loin avec votre PC.

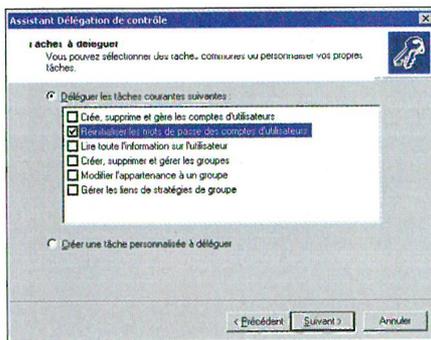
Compatibles **PCmagazine** : le magazine des utilisateurs avancés

ACTUELLEMENT EN VENTE CHEZ VOTRE MARCHAND DE JOURNAUX



Attention: dans le cas d'un poste connecté en réseau, les options de sécurité du domaine ont priorité sur les options de sécurité locales en cas de conflit.

En effet, dans Active Directory, il est possible (et conseillé) de définir des stratégies de groupe, qui peuvent s'appliquer à tout conteneur Active Directory: sites, domaines et unités organisationnelles (OU).



> Écran 5 : la délégation d'administration.

Les stratégies de groupe (W2k) uniquement)

Les stratégies de groupe affectent tous les utilisateurs et tous les ordinateurs du conteneur auxquelles elles s'appliquent, et peuvent être contrôlées ensuite par les groupes auxquels appartient l'utilisateur et l'ordinateur.

L'ordre d'application des stratégies de groupe est le suivant :

- Stratégies style NT 4.0 (NTConfig.pol).
- Stratégie de groupe locale.
- Stratégies de site, dans l'ordre administratif.
- Stratégies de domaine, dans l'ordre administratif.
- Stratégies d'OU, du haut vers le bas et dans l'ordre administratif.

Par défaut, les dernières stratégies « écrasent » les premières. Il est donc extrêmement important de bien définir ces stratégies de groupe afin que les paramètres de sécurité soient appliqués dans le bon ordre. Pour créer une stratégie de groupe, ouvrez l'outil d'administration « **Utilisateurs et groupes Active Directory** », faites un clic droit sur le conteneur qui vous sied et choisissez **Propriétés**. Cliquez sur l'onglet **Stratégie de groupe** (voir l'écran 2).

Vous pouvez alors créer les stratégies de votre choix sur chacun des conteneurs de votre forêt Active Directory (voir l'écran 3).

Gestion des comptes utilisateurs

Sous Windows 2000, la gestion des comptes utilisateurs se fait par l'outil d'administration « **Gestion de l'ordinateur** » (nœud **Utilisateurs et groupes locaux**) pour les comptes définis dans la base SAM locale, ou par l'outil d'administration « **Utilisateurs et ordinateurs Active Directory** » pour les comptes définis dans Active Directory sur un contrôleur de domaine.

Il n'est pas inutile de prodiguer quelques recommandations. Utilisez des comptes distincts pour l'administration et l'activité de tous les jours des utilisateurs. Pour éviter toute modification accidentelle de ressources sen-

sibles, il est conseillé d'utiliser le compte ayant le moins de privilèges pour effectuer la tâche souhaitée. Les virus notamment peuvent causer beaucoup plus de dommages s'ils sont activés par l'intermédiaire d'un compte ayant des privilèges élevés.

Sous Windows 2000, utilisez la commande **RunAs** pour lancer des outils d'administration avec des privilèges plus élevés, par exemple.

Renommez le compte Administrateur en un compte moins évident. Ce compte très

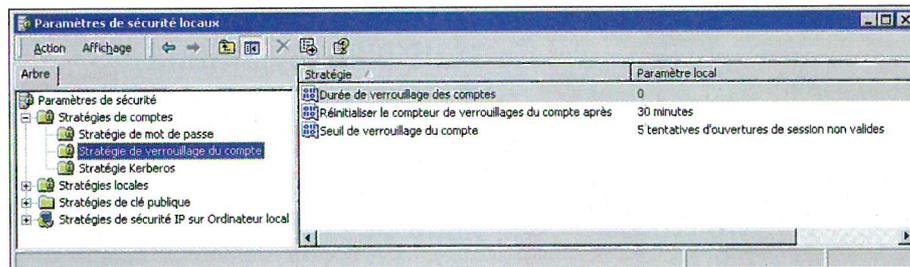
puissant est le seul qui ne peut être verrouillé à la suite de plusieurs échecs de tentatives d'ouverture de session. Les attaquants essaieront donc en priorité de pénétrer dans le système en tentant de deviner à plusieurs reprises le mot de passe de ce compte. En le renommant, vous faites perdre du temps supplémentaire aux attaquants. Vous pouvez de plus ajouter un compte leurre aux privilèges réduits nommé Administrateur si vous le désirez. Dans tous les cas, choisissez un mot de passe extrêmement fort pour le compte Administrateur réel.

Assignez un mot de passe au compte Invité et désactivez-le. Remarque : le compte Invité est déjà désactivé sur les versions Serveur de Windows NT 4.0 et Windows 2000.

Les bons gestes

Ouverture de session

Appuyer TOUJOURS sur **CTRL+ALT+SUPPR** avant d'ouvrir une session. En effet, des programmes conçus dans le but de recueillir des mots de passe peuvent apparaître comme un écran d'ouverture de session. En procédant de la sorte, vous désactivez ces programmes avant d'obtenir l'écran d'ouverture de session sécurisée de Windows NT.



> Écran 7 : la stratégie de verrouillage de comptes.

2 - Stratégie de verrouillage de comptes :

Seuil de verrouillage du compte	5.
Réinitialiser le compteur de verrouillage du compte après...	5 minutes minimum, plutôt 30 min à 1h.
Durée de verrouillage des comptes	Toujours ou au minimum un jour.
L'utilisateur doit se connecter pour changer son mot de passe (NT 4.0 seulement)	Oui.

Sous Windows 2000 Professional, la nécessité d'appuyer sur ces touches n'est pas obligatoire par défaut. Il faut l'activer par l'intermédiaire de l'onglet « Avancé » du panneau de contrôle « Utilisateurs et mots de passe » (voir l'écran 4).

Enfin, l'ouverture de session sous Windows 2000 supporte en standard l'utilisation de *Smart Cards* (cartes à puce). Vous pouvez connecter un lecteur de cartes supporté à l'ordinateur et utiliser votre carte pour vous authentifier, sans avoir à saisir de mot de passe.

Verrouillage de la station de travail

Paramétrez les écrans de veille pour qu'il verrouillent automatiquement la session en cas de non-utilisation pendant une durée déterminée. Pour cela, utilisez l'option « Protégé par mot de passe » de votre écran de veille.

Délégation d'administration

Windows 2000 permet de déléguer l'administration des éléments d'Active Directory (voir l'écran 5).

Il est conseillé d'utiliser cette possibilité afin de simplifier la tâche des administrateurs et d'assurer une administration plus proche des ressources.

Stratégies de comptes

Les stratégies de comptes définies par défaut dans Windows NT 4.0 et Windows 2000 ne sont pas suffisantes. Paramétrez vos stratégies comme indiqué dans les tableaux 1 et 2 et sur les écrans 6 et 7

Gestion et robustesse des mots de passe

- Changez de mot de passe fréquemment et en cas de doute de compromission. Évitez de réutiliser les mêmes mots de passe, et n'assignez pas le même mot de passe à des comptes différents.
- Évitez d'utiliser des mots qui peuvent facilement être devinés ou des mots du dictionnaire. Choisissez par exemple une combinaison de lettres, de chiffres et d'autres caractères non alphanumériques.
- N'écrivez votre mot de passe nulle part. Choisissez-en un que vous seul pouvez mémoriser facilement.

Activez le verrouillage du compte administrateur sur accès distant : sous NT 4.0, par défaut, comme nous l'avons vu, le compte administrateur ne peut être verrouillé après l'échec de plusieurs tentatives d'ouverture de sessions. Toutefois, il est possible d'activer le verrouillage de ce compte après l'échec de tentatives de connexions distantes, par le réseau. Pour cela, exécutez l'outil **PASSPROP** du kit de ressources techniques de Windows NT 4.0 (syntaxe: **C:\passprop /adminlockout**). Le compte administrateur sera alors verrouillé après le nombre d'échecs spécifié dans la stratégie de verrouillage de comptes (si elle est activée). L'administrateur pourra toujours se loguer en local sur le serveur pour déverrouiller le compte.

Forcez l'utilisation de mots de passe complexes : pour obliger les utilisateurs à choisir des mots de passe robustes, installez **PASSFILT**. Il s'agit d'un filtre de mots de passe implémenté dans une DLL et fourni en standard avec NT 4.0 depuis le SP2.

Ce filtre impose les 3 règles suivantes :

- Le mot de passe ne doit pas contenir le nom de *login* ou un morceau du nom complet de l'utilisateur.
- Il doit faire au moins 6 caractères de long.
- Il doit contenir des caractères d'au moins 3 des 4 jeux suivants :
 - alphabétiques minuscules,
 - alphabétiques majuscules,
 - chiffres,
 - caractères non alphabétiques (\$,!,%,^, ...).

Pour installer **PASSFILT**, vérifiez que le fichier **PASSFILT.DLL** se trouve bien dans **\WINNT\system32** (il a du être installé par un Service Pack) ou copiez-le à la main, puis dans la clé suivante de la Registry : **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa**, ajoutez la chaîne « **passfilt** » (sans « .dll ») à la valeur « *Notification Packages* ». Si la valeur n'existe pas, créez-la.

Attention : un administrateur pourra toujours changer un mot de passe directement dans la SAM avec le gestionnaire d'utilisateurs sans que ce mot de passe ne transite par le filtre. Enfin, sous Windows 2000, le durcissement des mots de passe peut se configurer en standard de manière simple. Comme nous l'avons vu, dans les paramètres de sécurité locale, activez « *Password must meet complexity requirements* ».

– Chiffrez la base de comptes : la SAM de Windows NT 4.0 contient les *hashes* des mots de passe LanMan et NTLM des comptes. Pour éviter que ces *hashes* ne soient récupérés directement par des outils de « craquage » de mots de passe comme L0phtCrack, utilisez l'utilitaire SYSKEY pour chiffrer la SAM locale. SYSKEY vous offre le choix de demander un mot de passe au *boot* de Windows NT, ou de stocker la clé de chiffrement sur une disquette qu'il faudra fournir à chaque démarrage, ou bien encore de stocker la clé de chiffrement sur le disque dur. Si vous voulez que votre serveur puisse redémarrer seul, choisissez cette dernière option.

Sous Windows 2000, les *hashes* des mots de passe sont protégés dans la SAM par un mécanisme de salage (*salting*), et SYSKEY n'existe plus. Mais malgré ces mécanismes de protection, il existe des outils diffusés sur Internet qui permettent tout de même de récupérer ces *hashes*, aussi bien sur NT 4.0 que sur Windows 2000, sous certaines conditions.

Le mois prochain, nous verrons les droits utilisateurs, les permissions d'accès aux fichiers et répertoires, le chiffrement de fichiers et de répertoires, le paramétrage des clés de la base de registre, le paramétrage des permissions d'accès aux clés de la base de registre, l'activation de l'audit du système et le contrôle périodique de l'état de votre système. ■

L'auteur : Patrick Chambet (patrick.chambet@edelweb.fr) est expert en sécurité Windows NT et W2K au sein d'Edelweb (<http://www.edelweb.fr>), l'une des premières sociétés françaises de conseil en sécurité des systèmes d'information et qui comporte un pôle sécurité Windows 2000.

Cachez ce seing que je ne saurais voir



Que l'on soit juste prudent ou définitivement paranoïaque, dès que l'on approche d'Internet le chiffrement se révèle incontournable. Nous vous livrons un des pseudos de la plus performante barrière à la curiosité : RC6.

Christophe Babayou et Laurent Foynard

Depuis l'avènement d'Internet, la sécurisation réseau n'a cessé de prendre de l'importance. De nos jours c'est un passage obligé pour toute communication confidentielle sur la toile, dans un réseau local, voire même pour le stockage de données sensibles.

L'un des principaux composants de la sécurisation, est le cryptage. L'article que vous êtes en train de lire va vous permettre de concevoir dans n'importe quel langage de programmation, l'un des algorithmes de chiffrement le plus performant du moment, le RC6. (*)

Présentation

Le RC6 est un algorithme inventé par la société RSA Data Security Inc. (RSADSI). Il participa à un concours organisé par le NIST, qui avait pour but de trouver un remplaçant à l'archaïque DES, comme standard cryptographique. Le gagnant prendrait alors le doux nom de AES (*Advanced Encrypting Standard*). Bien qu'il fut battu en finale, il n'en reste pas moins un incroyable outil de sécurisation qui allie trois qualités indispensables, à savoir simplicité, rapidité et puissance.

Le RC6, comme l'ensemble des candidats AES, est un algorithme de chiffrement symétrique, c'est-à-dire qu'une même

clé sert au cryptage et au décryptage. Il est donc primordial de garder ce code à l'abri des regards indiscrets, car toute personne étant en possession de cette dernière sera capable de déchiffrer sans aucun problème tous ce que vous crypterez.

Selon le désir de l'utilisateur, la longueur de la clé de chiffrement peut aller jusqu'à 256 bits. Afin de respecter la législation française (voir l'encadré à ce sujet) nous ne vous donnerons les explications de fonctionnement que pour des clés de 128 bits maximum.

Le chiffrement s'effectue par blocs de 128 bits, c'est-à-dire que l'ensemble d'un texte n'est pas crypté en une seule fois, mais morceau par morceau, sans aucune limite de longueur. La taille des blocs étant fixe, il est nécessaire de rajouter des octets de bourrage dans le cas où le dernier bloc n'aurait pas une longueur satisfaisante. Cela est assuré par une technique appelée *padding*, que nous verrons en détail plus loin.

Contrairement à ce que l'on pourrait penser, de nombreux algorithmes de chiffrement peuvent crypter de différentes manières, ces dernières étant totalement incompatibles les unes avec les autres. Le RC6 ne fait pas exception à la règle, et est capable de crypter de quatre manières différentes (voir encadré). Nous expliquerons le plus simple d'entre eux, à savoir l'ECB.

La législation française

En France, la cryptologie, comme tout ce qui s'y rattache, est considérée comme une arme extrêmement dangereuse. Le cryptage destiné à un usage autre que militaire n'est d'ailleurs autorisé que depuis février 1998, et encore sous de nombreuses conditions.

Toute personne concevant des logiciels de sécurisation doit prévenir le gouvernement, et demander une autorisation de vente. Les clients de ces logiciels sont également soumis à autorisation. La seule entorse à la règle concerne les produits de démonstration, qui ne sont pas soumis à autorisation du moment que la clé de cryptage n'excède pas 128 bits.

Fonctionnement

Voici les différentes étapes de cryptage et de décryptage du RC6 en mode ECB.

Cryptage :

- Entrée du texte clair sous forme d'une série de blocs de 128 bits (8 octets) chacun.
- Ajout du *padding* dans le dernier bloc.
- Entrée de la clé de cryptage afin de générer différents tableaux nécessaires au cryptage.

* Note : dans la totalité de cet article, nous allons parler de texte à crypter. En réalité ce terme définit n'importe quel fichier, aussi bien du texte, qu'une image qu'un programme exécutable.

- Appel de la méthode de cryptage pour chaque bloc de texte, les uns après les autres.
- Concaténation des blocs cryptés afin de composer un texte qui est la version chiffrée du texte initial.

Décryptage :

- Entrée du texte crypté sous forme d'une série de blocs de 128 bits (8 octets) chacun.
- Entrée de la clé de chiffrement afin de générer différents tableaux nécessaires au décryptage.
- Appel de la méthode de décryptage pour chaque bloc de texte, les uns après les autres.
- Concaténation des blocs décryptés afin de composer un texte qui est la version décryptée du texte initial, avec le *padding*.
- On retire le *padding* afin d'obtenir un texte qui est la copie conforme du texte clair initial.

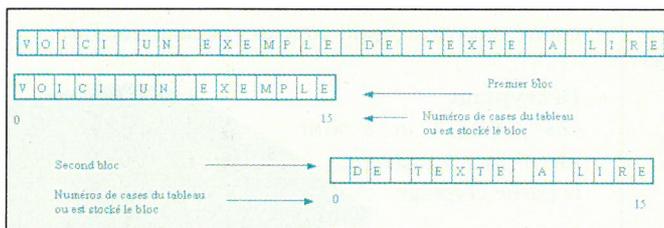
Maintenant que vous avez eu un aperçu un peu plus pointu de l'algorithme, nous allons pouvoir passer à la partie principale de cet article, à savoir le pseudo-code.

Le pseudo-code

Les explications sont découpées en étapes et suivent le plan établi ci-dessous. Bien entendu le pseudo-code, comme son nom l'indique n'est pas un programme clé en main, mais plutôt une sorte de plan de travail qu'il faudra adapter à votre langage de programmation préféré, en fonction de vos désirs et besoins.

Cryptage :

1 – La récupération du texte peut être faite de différentes manières. La plus simple et la plus générique est la lecture di-



> Écran 1 : dans cet exemple, nous avons considéré que nous regardons le contenu d'une case comme du code ascii, en réalité, une case contient la valeur hexadécimale de chaque lettre. En fait, l'algorithme n'interprétera jamais le contenu d'un bloc, car dans le cas d'un exécutable ou autre, le contenu d'une case n'est pas directement interprétable.

recte d'un fichier, mais on peut très bien imaginer que l'utilisateur souhaite écrire un message destiné à être crypté (pour un mail par exemple).

La découpe du texte clair en bloc de 128 bits se fait par l'intermédiaire d'un tableau d'octets de 16 cases. Comme le cryptage de chaque bloc est indépendant l'un de l'autre, vous pouvez soit utiliser ce seul tableau que vous réinitialiserez avec un autre bloc, soit un tableau à deux dimensions. La première solution est à mon sens la plus économique en termes d'espace mémoire et de rapidité.

Le stockage des données dans le tableau se fait de la manière suivante. Le tout premier octet du bloc sera stocké dans la case 0 du tableau, tandis que le dernier octet sera stocké

dans la case 15. Ce type de stockage s'appelle le « *Big Endian* », c'est la technique de stockage la plus fréquemment utilisée. Il est important de faire attention à cela, car de nombreux algorithmes de cryptage fonctionnent en *Big Endian* (voir l'encadré et l'écran 1 à ce sujet).

Le padding

2 – Pour que le RC6 fonctionne convenablement, il faut impérativement que tous ses blocs fassent exactement 16 octets. Comme il est rare qu'un texte génère un nombre d'octets multiple de 16, il faut employer une technique de bourrage (*padding* en anglais), afin d'atteindre une taille acceptable. Cela ne se fait bien entendu pas n'importe comment. La technique présentée ici (employée par tous les algorithmes concourant pour devenir l'AES) est directement inspirée de celle du DES. Voilà comment il convient de procéder : vous comptez le nombre d'octets qui manquent à votre dernier bloc pour être complet, et vous le complétez avec des octets qui ont pour valeur, le nombre décimal représentant le nombre d'octets manquants (voir l'écran 2).

Si par hasard il s'avère que votre texte fait exactement le nombre d'octets nécessaire, il convient de rajouter tout un bloc supplémentaire, dont chaque octet aura la valeur 16, en décimal. De ce fait, lorsque l'on souhaitera retirer le *padding* (voir plus loin) il suffira de lire le dernier octet du dernier bloc pour connaître le nombre d'octets non significatifs.

3 – La clé de cryptage (qui rappelons-le sert également au décryptage) peut être choisie par l'utilisateur ou bien générée aléatoirement, mais doit de toute manière être connue de la personne qui crypte et de la personne qui devra décrypter. Elle va servir à produire un bloc de 44 sous-clés de 32 bits chacune. Ce sont elles qui seront utilisées lors du cryptage et du décryptage en lieu et place de la clé originelle.

La clé est séparée en 4 registres de 32 bits chacun $L[0 \dots 4]$. La méthode de remplissage des registres respecte la norme *Big Endian* (voir page 85).

Elle sera ensuite entrée dans un module qui générera un tableau de 44 mots de 32 bits $S[0 \dots 43]$.

Les mots P32 et Q32 contiennent respectivement les valeurs hexadécimales suivantes : **B7E15163** et **9E3779B9**.

Notez que toutes les opérations ci-dessous sont effectuées sur des blocs de 32 bits.

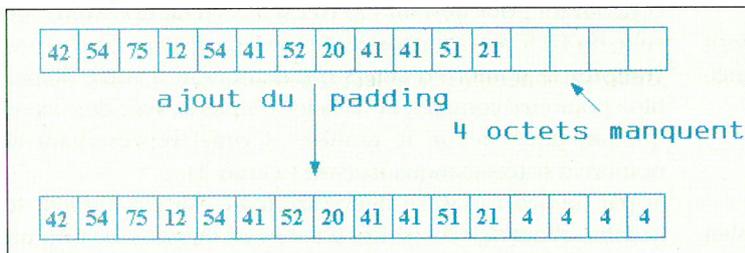
Voici le pseudo-code du module :

```
S[0] = P32 ;
For i=1 to 43
{
  S[i] = S[i-1] + Q32 ;
}
A = B = i = j = 0 ;
For s=1 to 132
{
  A = S[i] = ( S[i] + A + B ) RLC 3 ;
  B = L[j] = ( L[j] + A + B ) RLC ( A + B ) ;
  i = ( i + 1 ) % 44 ;
  j = ( j + 1 ) % 4 ;
}
}
```

Le symbole « RLC N » représente une rotation circulaire de n bits à gauche. C'est-à-dire que tous les bits du mot de 32 bits sont décalés n fois vers la gauche, et que tous les bits « sortant » du mot sont réinjectés à droite.

4 – La partie chiffrement à proprement parler peut maintenant commencer. Le module ci-dessous devra être effectué sur chaque bloc de texte clair, indépendamment les uns des autres.

Avant toutes choses, il faut séparer les 128 bits du bloc de texte clair, en 4 variables de 32 bits chacune, respectivement nommé A, B, C et D. Le tout en respectant une fois de plus la norme *Big Endian* (la variable A contiendra les 32 bits de gauche du bloc de texte clair).



> Écran 2 : voici un exemple d'ajout de padding dans le dernier bloc d'un texte.

Les différents modes de chiffrement

1. Le mode ECB : « *Electronic Code Book* = carnet de codage électronique » (codage par bloc). Dans ce mode, il est possible de concevoir un carnet de codage, c'est-à-dire qu'à un bloc de texte crypté, correspond un bloc de texte clair. Ceci présente un avantage : les blocs étant chiffrés et déchiffrés indépendamment, il devient possible de paralléliser les opérations.

2. Le mode CBC : « *cipher block chaining* = chiffrement avec chaînage par bloc » (codage par bloc). Ce mode utilise une rétroaction, le bloc suivant est codé en fonction de la sortie du bloc précédent. Le texte clair est combiné avec un « OU exclusif » avec le bloc chiffré précédent avant d'être codé. Pour ce mode, il faut initialiser un bloc de départ afin que la rétroaction soit possible avec le premier bloc de données.

3. Le mode CFB : « *cipher block feedback* = chiffrement à rétroaction » (codage en continu). Les données peuvent être traitées par unités plus petites qu'un bloc. On peut choisir la taille du tampon. Comme il y a une rétroaction, il faut initialiser une valeur de départ.

4. Le mode OFB : « *Output Feedback* = rétroaction de sortie » (codage en continu). Cette méthode utilise un algorithme de chiffrement par bloc comme un algorithme de chiffrement en continu. Similaire au chiffrement CFB. Le processus de chiffrement est exactement l'inverse du chiffrement. Il faut un vecteur d'initialisation.

En sortie du module, ces 4 mêmes variables contiennent des morceaux du texte chiffré, qu'il faudra concaténer pour former un bloc crypté, correspondant au bloc de texte entré. Voici le pseudo-code :

```
B = B + S[0];
D = D + S[1];
For i=1 to 20
{
  t = ( B * ( 2*B+1 ) ) RLC 5;
  u = ( D * ( 2*D+1 ) ) RLC 5;
  A = ( ( A xor t ) RLC u ) + S[2*i];
  C = ( ( C xor u ) RLC t ) + S[2*i+1];

  (A,B,C,D) = (B,C,D,A);
}
A = A + S[42];
C = C + S[43];
```

La ligne de pseudo-code (A,B,C,D) = (B,C,D,A) signifie que les blocs échangent leurs valeurs d'un cran vers la gauche. A prendra la valeur de B, tandis que B prendra celle de C, etc. Cette opération peut être réalisée en utilisant le code suivant, qui fait intervenir une variable intermédiaire qui est, elle aussi, un mot de 32 bits.

```
Intermédiaire=A
A=B
B=C
C=D
D= intermédiaire
```

5 – Il suffit maintenant de concaténer les blocs les uns à la suite des autres (le premier bloc crypté en premier, suivi du second, etc.), afin d'obtenir un texte chiffré qui correspond au texte initialement entré.

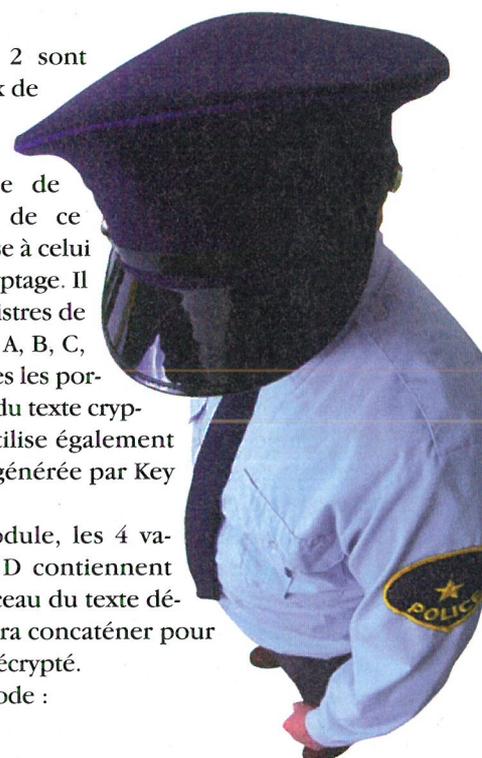
Déchiffrage

Les points 1 et 2 sont identiques à ceux de la partie cryptage.

3 – Le principe de fonctionnement de ce module est inverse à celui du module de cryptage. Il utilise aussi 4 registres de 32 bits en entrée A, B, C, D où sont stockées les portions de 128 bits du texte crypté. Ce module utilise également la clé S[0 ... 43] générée par Key Schedule.

En sortie du module, les 4 variables A,B,C et D contiennent chacune un morceau du texte déchiffré, qu'il faudra concaténer pour former un bloc déchiffré.

Voici le pseudo-code :



Cachez se seing que je ne saurais voir

```
C = C - S[2*r+3];
A = A - S[2*r+2];
For i=20 downto 1
{
  (A,B,C,D) = (D,A,B,C);
  u = ( D*( 2*D+1 ) ) RLC 5;
  t = ( B*( 2*B+1 ) ) RLC 5;
  C = (( C - S[2*i+1] ) RLC t ) xor u;
  A = (( A - S[2*i] ) RLC u ) xor t;
}
D = D - S[1];
B = B - S[0];
```

4 – Il convient maintenant de concaténer les blocs les uns à la suite des autres (toujours en suivant la norme *Big Endian*, le premier bloc crypté en premier, suivi du second, etc.), afin d'obtenir un texte clair, suivi du *padding* qui avait été ajouté avant cryptage.

5 – Retirer le *padding* est une opération simplissime, puisqu'il suffit de lire la valeur décimale du tout dernier octet du texte obtenue dans la partie 4. La valeur de cet octet indique le nombre d'octets à retirer de la fin du texte, afin d'obtenir le texte clair initialement entré.

Une fois ces quelques octets retirés, vous pouvez soit sauvegarder ce texte dans un fichier qui devra de préférence être du même type qu'au départ (par

exemple, dans le cas d'un fichier exécutable, il devra être suivi de l'extension « .exe » sous Windows ou précisé exécutable sous Linux).

Conclusion

Vous avez maintenant toutes les cartes en main pour réaliser dans de parfaites conditions un logiciel qui exploitera le RC6. Il nous reste cependant un dernier conseil à vous donner, lors de la réalisation de vos programmes, faites très attention au cryptage et au décryptage du bloc qui contient le *padding*. Si jamais il y a une erreur de cryptage/décryptage sur le dernier octet du *padding*, sa valeur peut ne pas correspondre au nombre d'octets réels du texte clair, et ainsi provoquer de nombreux problèmes. Faites également attention lorsque vous cryptez/décryptez autre chose que des fichiers texte, car le moindre octet en plus ou en moins peut provoquer un dysfonctionnement complet du fichier concerné. ■

Little Endian/Big Endian Le *Big* et le *Little Endian* décrivent deux modes de stockage de données. En *Little Endian*, les octets les plus significatifs sont placés en premier. Alors qu'en *Big Endian*, ce sont les octets les moins significatifs qui sont codés en premier.

NOUVEAU

monweb

WWW.

Mettez-vous à la p@ge !

monweb vous accompagne dans la réalisation de vos premiers sites, vous informe sur les derniers logiciels pour mieux exploiter et maîtriser Internet, et vous guide dans vos projets personnels ou d'entreprise

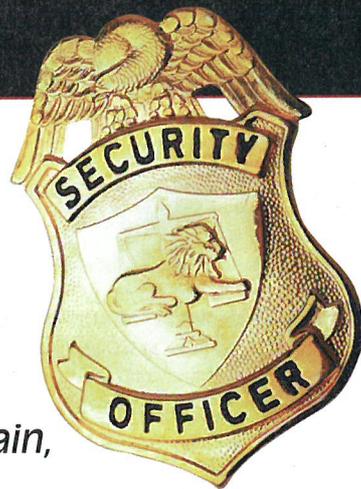
LE MAGAZINE
DU
WEBCRÉATEUR

Sur le CD-ROM du magazine, tous les outils nécessaires aux réalisations pratiques présentées dans le magazine (versions complètes ou d'essai)



EN VENTE ACTUELLEMENT CHEZ VOTRE MARCHAND DE JOURNAUX

X509, l'e-norme



Comment déterminer si le composant ActiveX que vous venez de télécharger est authentique? Pour en être certain, vous devez disposer d'un certificat d'authenticité.

Par Xavier Leclercq
xavier.leclercq@programmez.com

Microsoft s'est basé sur la norme X509 pour que l'industrie du *on-line*, englobant le développement du commerce et des transactions, puisse élaborer des solutions adéquates à ce problème de sécurité. Et si le nombre de certificats actuellement en circulation n'est que de quelques dizaines de milliers, ils devraient à brève échéance se compter bientôt en dizaines de millions. Un vrai défi pour le marché du commerce électronique qui se chiffrera à plusieurs milliards de dollars à l'horizon 2005 !

Pour résoudre la problématique du téléchargement et de l'installation de composants ActiveX à travers une liaison HTTP, Microsoft a proposé sa solution propriétaire qui porte le nom de format CAB (« cabinet »). Celui-ci comprime une série de fichiers grâce à une variante publique de l'algorithme de compression Lempel-Ziv. Pour emballer le tout, la société de Seattle a aussi prévu une couche de sécurité, visant à authentifier la source serveur pour déterminer sa fiabilité. À partir du moment où le chargement d'une application Java nécessite autant d'appels à HTTP qu'il y a de classes, et à l'heure où l'on parle d'engorgement du réseau, une telle solution logicielle s'avère indispensable.

Nous allons parcourir ensemble un résumé non exhaustif de la théorie et de la pratique de l'authentification.

Qu'est que l'authentification ?

L'autorité de certification (*Certificate Authority*) – nous précisons plus loin de quoi il s'agit – choisie par Microsoft est la société VeriSign. Le site <http://digitalid.verisign.com> précise qu'une signature électronique est fournie gratuitement (mais pour une période limitée de 60 jours) aux utilisateurs qui en font la demande (voir l'écran 1).

Sous Windows 9x ou NT 4 / Windows 2000, vous pouvez personnaliser à volonté l'aspect de votre bureau ou encore employer des ressources partagées (imprimantes, CD-ROM, unités de sauvegarde). Au démarrage du système d'exploitation, l'ordinateur associera vos préférences à votre « *UserName* ». Et si l'ordinateur a confiance en votre identité c'est parce que vous lui avez précisé un mot de passe que vous seul (en théorie) détenez. Cette action de vérification d'identité d'un individu se nomme « processus d'authentification ».

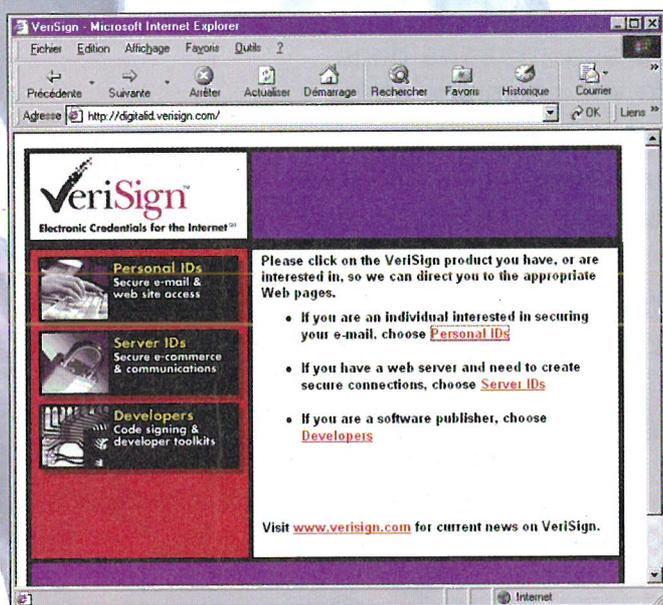
L'authentification simple

Il y a de nombreuses années, les responsables de la sécurité se sont très vite rendu compte que la simple association du couple « nom d'utilisateur / mot de passe » dans un fichier centralisé était trop vulnérable. Bien entendu, on a pensé à coder les mots de passe et même à les fragmenter. Néanmoins, la présence de ce fichier de mot de passe représente un « trou » potentiel de sécurité.

L'ordinateur a-t-il vraiment besoin de stocker les mots de passe? Non : il suffit qu'il puisse distinguer un mot de passe correct d'un mot de passe incorrect. Il stockera non plus un mot de passe mais le résultat d'une fonction à sens unique.

- L'utilisateur se connecte en encodant son mot de passe.
- L'ordinateur applique au mot de passe la fonction à sens unique.
- Le résultat est comparé à celui qu'il détient dans sa base de données.

En théorie, un pirate ne peut plus rien faire avec le fichier de mots de passe, car on ne peut appliquer une fonction in-



Écran 1 : Une signature électronique gratuite pendant 60 jours.

verse pour le retrouver. Autrement dit, même si la fonction à sens unique est rendue publique, cela n'avancera en rien celui qui possède le mot de passe crypté.

En pratique une attaque est possible. À l'aide de la fonction à sens unique, il suffit de créer un fichier (appelé «dictionnaire») représentant la liste des mots de passe les plus utilisés. Les comparaisons avec les mots de passe chiffrés ont alors une bonne chance d'aboutir.

La parade à une attaque par dictionnaire (sous Unix / Linux) consiste à «saler» le mot de passe initial en injectant une chaîne supplémentaire aléatoire de x bits. Mais des études ont prouvé que dans plus de 20 % des cas, le mot de passe est vulnérable à une attaque par dictionnaire même avec un «sel» important.

L'authentification avec clé publique

Même en salant énormément, d'autres problèmes se posent.

- Comment pouvez-vous être certain que vous vous adressez au bon serveur (hôte distant) ?
- Imaginons un schéma d'authentification simple qui s'appliquerait à un ordinateur distant. Entre vous et l'hôte distant existe un chemin d'accès. Le risque étant que toute personne ayant un accès aux ordinateurs intermédiaires pourrait lire votre mot de passe.

La solution au deuxième problème est de ne jamais communiquer son mot de passe confidentiel (sa clé privée). L'ordinateur hôte ne connaîtra qu'une clé publique.

- Le host envoie une chaîne aléatoire quelconque à l'utilisateur.
- À l'aide de sa clé privée l'utilisateur code le message et le renvoie à l'ordinateur distant.
- Celui-ci le décode à l'aide de sa clé publique. Si le message envoyé initialement correspond bien au message décodé, l'utilisateur hôte peut certifier que l'utilisateur est bien celui dont il possède les coordonnées dans sa base de données.

La signature électronique

Monsieur A et Monsieur B désirent s'échanger des informations confidentielles. Un script VB génère à la demande de Monsieur B deux clés :

- La première est une clé de chiffrement et est envoyée à Monsieur A. Elle est dite «publique».
- La deuxième est gardée secrète par Monsieur B. Monsieur A chiffre son message à l'aide de la clé publique spécifique au destinataire et lui envoie. Monsieur B est alors le seul à pouvoir décoder le message à l'aide de sa clé privée tenue secrète. Si le message codé et la clé publique sont interceptés par une tierce personne, celle-ci est incapable de reconstituer le message. Même Monsieur A ne peut plus reconstituer le message d'origine avec sa seule clé publique.

Pourquoi la clé est-elle «publique»? Vous pouvez légitimement penser qu'il existe un risque qu'une clé publique puis-

se générer à son tour un message falsifié. En fait, il est possible de contrer cette possibilité «en signant» chaque message. L'expéditeur (Monsieur A) possède lui aussi une clé publique et une clé privée. Il envoie à Monsieur B sa clé publique et chiffre un message avec sa clé secrète (il «signe» ainsi le message). Puis, l'ensemble du message est à son tour crypté à l'aide de la clé publique de Monsieur B. À la réception du message ce dernier emploie sa clé privée pour le déchiffrer, puis, lui applique la clé publique de Monsieur A pour ainsi certifier la validité de ces informations !

Les choix de Microsoft : les environnements d'authentification X.509 et PKCS#10

X.509 se base sur la notion de certificat. Le but recherché n'étant plus «de dialoguer» de manière confidentielle mais de s'assurer de l'intégrité de ce qui a été chargé *via* le réseau. De quoi s'agit-il ? Imaginons qu'un «éditeur» au sens large désire publier un logiciel (il peut s'agir aussi bien de l'auteur d'un «shareware» que d'un éditeur professionnel). Il soumet sa demande à une Autorité de Certification (AC) en lui fournissant une clé publique. L'éditeur génère une double clé. Une clé privée qu'il conservera précieusement (qui pourra être sauvegardée *via* un mécanisme *hardware* spécialisé) et une clé publique. Cette clé publique est au format PKCS#10 (*Public-Key Cryptography Standards* : norme de la société RSA) et est codée «base64».

L'AC accepte ou non sa demande de certification et dans un délai d'une semaine (dans le cas de VeriSign), transmet à l'éditeur son certificat. Une copie du certificat est bien entendu conservée sur le serveur de l'AC. Celle transmise à l'éditeur contient une clé publique. Pour rappel, chaque certificat est «signé». L'AC possède lui aussi une clé publique et une clé privée. Il envoie à l'éditeur sa clé publique et chiffre un message avec sa clé secrète (il «signe» ainsi le message). Puis l'ensemble du message est à son tour crypté à l'aide de la clé publique de l'éditeur. À la réception du message, ce dernier emploie sa clé privée pour le déchiffrer puis lui applique la clé publique de l'AC pour ainsi certifier l'origine du certificat.

L'éditeur signe un objet qu'il a conçu : un contrôle ActiveX bien entendu, mais aussi une *applet* Java, un *plug-in*, un fichier CAB, un exécutable .exe, etc. Cet objet se nomme un PE (portable-executable).

Le codage de l'intégralité du PE serait un processus beaucoup trop lent. Et c'est la principale raison qui pousse encore l'emploi d'un mécanisme d'authentification simple à clé privée. En effet, l'authentification avec clé publique se révèle jusqu'à 1 000 fois plus lente !

Bref, Microsoft a décidé de «couper la poire en deux». Dans un premier temps le PE est divisé en blocs de x octets. À chaque bloc un algorithme applique une fonction de hachage unique qui en retour fournit une somme de contrôle. Il s'agit en quelque sorte d'un «checksum» du même genre que celui du secteur *boot* de démarrage. Si un octet est modifié, la somme de contrôle n'est plus la

même et l'utilisateur peut en déduire que le bloc a été modifié (voir la **figure 1**).

La somme de contrôle, le «*digest number*», est ensuite codée à l'aide de la clé privée de l'éditeur. Le PE contient donc le code d'origine, la clé publique, le certificat avec les valeurs cryptées de la fonction *hash* de chaque bloc.

Concrètement :

- Voici un bloc contenant la chaîne «test signature» au format Unicode (un caractère occupe 16 bits) :

```
0x00000000 54 00 65 00 73 00 74 00 T.e.s.t.
0x00000008 20 00 53 00 69 00 67 00 .S.i.g.
0x00000010 6e 00 61 00 74 00 75 00 n.a.t.u.
0x00000018 72 00 65 00 r.e.
```

Étape 1 du schéma.

- Voici la somme de contrôle, résultat de la fonction *hash* appliquée à ce bloc :

```
0x00000000 a8 2b df c2 c9 f1 bb 62 .+....b
0x00000008 38 78 d4 60 fa ce 5c 2a 8x.'..\'*
```

Étape 2 du schéma.

- La somme de contrôle est emballée dans une structure spéciale du nom de PKCS#1.

Cette structure contient la somme de contrôle, mais aussi le nom de l'algorithme de la fonction *hash* (qui n'est pas fixé une bonne fois pour toute).

```
0x00000000 2a 5c ce fa 60 d4 78 38 *...`x8
0x00000008 62 bb f1 c9 c2 df 2b a8 b.....+
0x00000010 10 04 00 05 05 02 0d f7 .....
0x00000018 86 48 86 2a 08 06 0c 30 .H.*...0
0x00000020 20 30 00 ff ff ff ff 0.....
0x00000028 ff ff ff ff ff ff .....
0x00000030 ff ff ff ff ff ff .....
0x00000038 ff ff ff ff ff 01 00 .....
```

Étape 3 du schéma.

- Enfin la clé privée est appliquée à la structure PKCS#1 :

```
WinVerifyTrust(
    HWND hwnd,
    DWORD dwTrustProvider,
    DWORD dwActionID,
    LPVOID ActionData,
    );
; Le service API WinVerifyTrust() vérifie l'intégrité
; du P.E. qui a été chargé.
```

```
0x00000000 64 f6 46 3a 97 2e 83 38 d.F...8
0x00000008 09 57 43 cb ca 41 59 0d .WC..AY.
0x00000010 03 35 d6 e4 36 6f 2c fc .5..6o,.
0x00000018 63 43 95 c4 fd e2 c4 ed cC.....
0x00000020 06 da 9a 21 98 fc 0a 6e ....l...n
0x00000028 3f 1c ad 3a db 7c 83 2f ?...l./
0x00000030 14 d0 58 80 02 df dc 96 ..X....
0x00000038 70 09 00 e3 5b bd 2b 45 p...[.+E
```

Étape 4 du schéma.

En C++ cela donne à peu près ce que l'on a sur le **listing 1**.

- Un utilisateur navigue sur le Web et tombe à un moment donné sur une page HTML qui référence le PE. Automatiquement le PE sera téléchargé pour être installé sur le poste client. Puis MSIE appellera la fonction API **WinVerifyTrust()**. Celle-ci extrait le certificat du PE et affiche la date d'expiration de celui-ci, le nom de l'éditeur et le nom du CA. La clé publique est extraite et un double traitement est pratiqué comme le montre le schéma ci-dessous. La fonction *hash* est à nouveau appliquée sur le code du PE. Chaque nombre obtenu est comparé au résultat décrypté à l'aide de la clé publique. Si la somme de contrôle décryptée ne correspond pas à celle nouvellement obtenue par l'application de la fonction hash, c'est qu'alors le PE a été modifié (virus, falsification, PE endommagé, ...) ou que la clé publique n'est pas correcte. Quoi qu'il en soit, l'utilisateur en est alors averti.

Astuce de Microsoft : en «hachant» le code, il est possible de l'exporter. En effet, le PE en lui-même n'est pas crypté!

La signature digitale employée par Microsoft a une longueur de clé de 1024 bits. Ce qui signifie qu'il faudrait une puissance de 90 milliards de MIPS/année pour réussir à en «casser» le code. Cette puissance nécessite la monopolisation durant 90 ans d'un milliard d'ordinateurs calculant à la vitesse d'un million d'instructions par seconde... Voir la **figure 2**

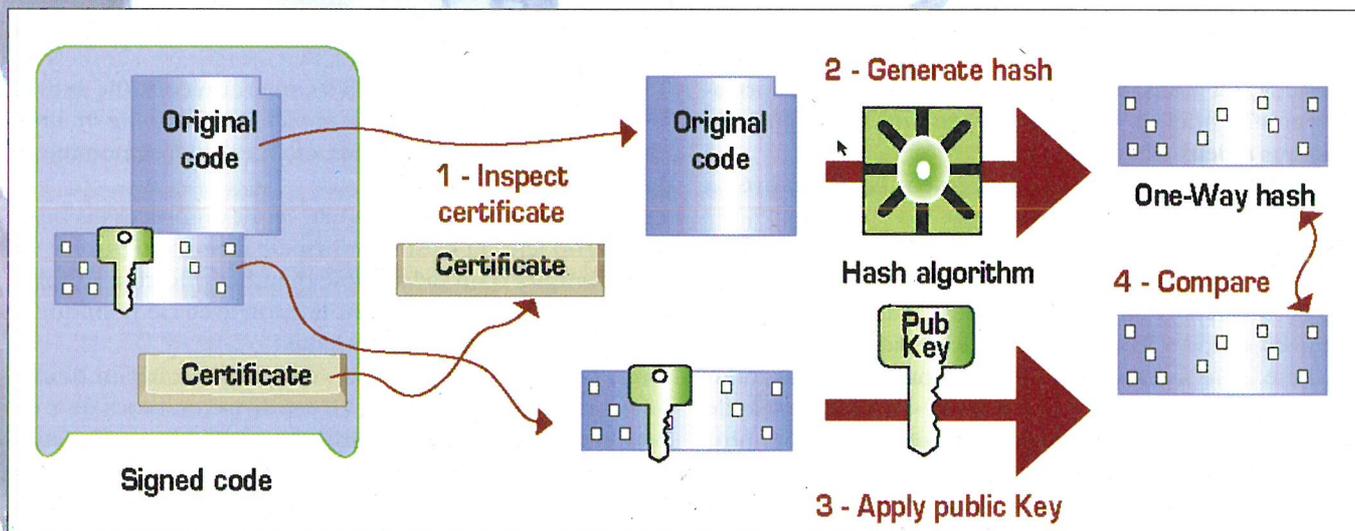


Figure 1 : le processus d'authentification.

Le protocole X.509 est un environnement d'authentification ISO qui peut fonctionner à travers plusieurs réseaux comme on peut le constater sur le schéma ci-dessous. Au sommet, il

ya une AC maîtresse. La structure est arborescente. Des AC différentes certifient d'autres AC, et d'autres AC certifient d'autres utilisateurs. Le «*root AC*», représente «le point de confiance commun».

Vous pouvez décider si l'éditeur est digne de confiance et l'ajouter à votre liste de confiance (le certificat d'un autre produit de l'éditeur ne sera plus systématiquement présenté. Mais en cas de falsification vous en serez toujours averti).

En termes de coût, en principe, une société comme VeriSign réclame 400 \$ par an si la demande émane d'un éditeur commercial (classe 3). Pour un individu isolé le prix est fixé à 20 \$ par an (classe 2). Enfin, il est également possible d'obtenir un certificat gratuitement pour une période limitée de 60 jours.

Conclusion

La prochaine étape sera sans doute l'authentification dynamique. Celle-ci pousse plus loin le concept en proposant une authentification bilatérale de l'ordinateur hôte et de l'utilisateur qui se connecte au poste client.

Cette technique fournit une clé privée associée à l'utilisateur et différente pour chaque session. Ainsi on ne vérifie plus seulement l'authenticité du poste client mais aussi l'identité de celui qui s'y connecte. De plus, la clé privée devient moins confidentielle. Si elle est dérobée, on ne pourra rien en faire car à chaque nouvelle session l'ordinateur réclame un mot de passe «dynamique» différent !

Quoi qu'il en soit, la mise en place de clés publiques de chiffrement marque une avancée notable dans la sécurisation des accès au réseau Internet. Une série de fournisseurs (Microsoft, RSA, IBM, Netscape, Intel, Verisign, Entrust Technologies) croisent actuellement le fer pour imposer leur implémentation spécifique, ce qui ne facilite pas la mise en place d'une norme.

Bien naïf serait l'utilisateur qui penserait que tous les acteurs des réseaux œuvrent dans un but commun de standardisation et d'harmonie. La situation est paradoxale : les sociétés commerciales s'allient pour créer des standards puis se font concurrence en proposant leurs extensions spécifiques... ■

```
#include <wincrypt.h>

HCRYPTPROV hProv = 0;
#define BUFFER_SIZE 256
BYTE pbBuffer[BUFFER_SIZE];
HCRYPTHASH hHash = 0;
BYTE *pbSignature = NULL;
DWORD dwSigLen;
LPTSTR szDescription = TEXT("Test Data");
DWORD i;

// Get handle to the default provider.
if(!CryptAcquireContext(&hProv, NULL, NULL, PROV_RSA_FULL, 0)) {
    printf("Error %x during CryptAcquireContext!\n", GetLastError());
    goto done;
}

// Fill buffer with test data.
for(i = 0 ; i < BUFFER_SIZE ; i++) {
    pbBuffer[i] = (BYTE)i;
}

// Create hash object.
if(!CryptCreateHash(hProv, CALG_MD5, 0, 0, &hHash)) {
    printf("Error %x during CryptCreateHash!\n", GetLastError());
    goto done;
}
// Hash buffer.
if(!CryptHashData(hHash, pbBuffer, BUFFER_SIZE, 0)) {
    printf("Error %x during CryptHashData!\n", GetLastError());
    goto done;
}

// Determine size of signature and allocate memory.
dwSigLen = 0;
if(!CryptSignHash(hHash, AT_SIGNATURE, TEXT("«»), 0, NULL, &dwSigLen)) {
    printf("Error %x during CryptSignHash!\n", GetLastError());
    if(GetLastError() != NTE_BAD_LEN) goto done;
}
if((pbSignature = malloc(dwSigLen)) == NULL) {
    printf("Out of memory!\n");
    goto done;
}

// Sign hash object.
if(!CryptSignHash(hHash, AT_SIGNATURE, szDescription, 0, pbSignature, &dwSigLen)) {
    printf("Error %x during CryptSignHash!\n", GetLastError());
    goto done;
}

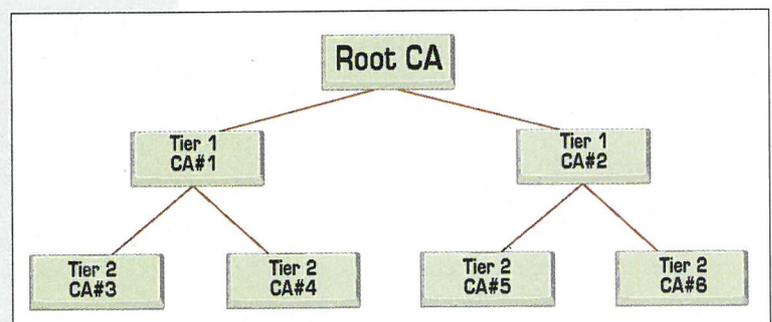
// Store or transmit the signature, test buffer, and description string.
...

done:

// Free memory used to store signature.
if(pbSignature != NULL) free(pbSignature);

// Destroy hash object.
if(hHash != 0) CryptDestroyHash(hHash);

// Release provider handle.
if(hProv != 0) CryptReleaseContext(hProv, 0);
```



> Figure 2 : savoir à qui faire confiance.

Laurent Ribardière, la passion du code

Laurent Ribardière fait partie des « grands » de la micro-informatique. À 24 ans il crée ACI devenue – sous le nom de 4D – une multinationale employant près de 180 personnes. Bien que président du groupe, Laurent conserve toujours la même passion pour la programmation.

Arnaud de la Pommeraye

Nous sommes au tout début des années 80, époque où naît la micro-informatique, le Z80, le PET Commodore, l'IBM PC, l'Apple II et le Lisa... C'est son prof de maths de seconde qui va le plonger dans l'univers de la programmation. « *Pour assouvir ma passion, j'allais le week-end au Palais de la Découverte et le reste de la semaine au Printemps qui possédait un rayon micro-informatique mélangé aux télévisions et à la Hi-Fi. Le chargé du rayon, très sympa et nous laissait programmer* », se souvient Laurent Ribardière. Le Basic fut le premier langage qu'il apprit, puis, très

acheté un Apple II pour mon passage en première. Pendant les grandes vacances, je suis retourné au Printemps pour retrouver mes copains et leur dire que j'avais un Apple II. Là, tout a changé, car le Printemps avait un accord avec une petite société d'informatique qui vendait du matériel. Je les ai rencontrés et ils m'ont proposé de faire des développements. Je travaillais le soir, après les cours, le samedi et le dimanche. J'ai collaboré avec eux de la première à Maths Spé. C'est en Maths Spé que j'ai pris la décision d'arrêter mes études pour me lancer dans l'informatique. À 24 ans, j'ai créé ACI.

machine. Le premier logiciel s'est appelé « ABC Base », en 1984, un petit gestionnaire mono-fichier qui a bien marché et qui nous a procuré suffisamment de moyens pour me laisser le temps de concevoir 4D. Le développement d'ABC Base a pris deux mois et demi pendant les grandes vacances ; pour 4D j'ai mis un an. J'ai programmé avec le Pascal d'Apple, un Pascal amélioré offrant, comme le C, un grand contrôle de la machine. Pendant des années, nous avons gardé cette configuration.

« L'arrivée du Lisa m'a convaincu »

rapidement, l'assembleur du 6502 afin de mieux maîtriser les actions du processeur. Puis ce fut au tour du Pascal UCSB toujours sur l'Apple II du Printemps alors dopé en mémoire par ajout d'une carte de 16 Ko. Parallèlement, son prof de maths l'initiait à l'algorithmique.

Programmez : *À cette époque, aviez-vous déjà le sentiment que vous feriez cela plus tard ?*

Laurent Ribardière : Pas complètement. Programmer était une véritable passion, un plaisir. Ma vocation est venue lorsque mes parents m'ont

Vous vous destinez à faire une carrière scientifique...

• Mon père rêvait de me voir rentrer à Polytechnique. J'ai arrêté sur un coup de tête : un jour, je me suis disputé avec l'un de mes professeurs pour une histoire d'horaires et il m'a dit « *si vous n'êtes pas content, ne revenez pas !* ». Je lui ai dit que je ne reviendrais pas. Mes parents m'ont soutenu parce que j'étais travailleur.

Vous vous lancez alors dans l'aventure ACI...

• Ce qui m'a convaincu, c'est l'arrivée en 1983 de Lisa, l'ancêtre du Macintosh. Il s'est avéré que Lisa n'a pas survécu... Mais je croyais au concept. J'ai commencé à développer pour cette

Vous aviez fait le choix de l'environnement Macintosh. Pourquoi avoir délaissé le PC ?

• Les notions de fenêtres et de souris m'ont séduit. C'était une philosophie de programmation très différente. Maintenant cela fait sourire puisque tout le monde est habitué à programmer en environnement graphique. À l'époque, Apple avait un slogan : « *tout ce que vous avez appris jusqu'à présent, oubliez-le et recommencez à zéro* ». Ce n'était pas vrai, mais il y avait un fond de vérité.

Vous parliez du plaisir du code rencontré à l'adolescence. L'avez-vous toujours ?

• Toujours. Je programme encore, mais pas autant qu'avant. Je m'amuse à programmer tant dans des langages comme le C++, que dans d'autres plus évolués comme 4D. Ce dernier a tellement évolué qu'on peut faire des

choses inimaginables il y a dix ans. Quand nous sommes en période de sortie de produit, tous les ingénieurs disponibles sont utiles et je les aide.

Qu'est-ce qu'un bon développeur ?

• Au début, nous étions une bande de jeunes. Certains disaient : « moi, je connais par cœur toutes les instructions du 6502 » et les autres étaient en admiration et pensaient « ce doit être un bon programmeur... » Pour moi, un bon développeur, c'est avant tout quelqu'un qui s'accroche, qui ne laisse pas tomber en cours de route. Autre critère : le choix du bon outil de travail. Le développeur doit être capable de choisir le langage le plus approprié à son projet et dépasser ses limites. Si on fait une application qui nécessite d'être rapide, il est préférable de choisir le C ou l'assembleur et non le Java. Enfin, il faut un esprit de synthèse.

Vous utilisez ces critères lors de l'embauche de programmeurs ?

• Absolument ! Nous recherchons aussi une certaine créativité et un esprit d'équipe. Lorsque l'on conçoit des solutions largement diffusées, il est impossible de développer seul dans son coin, car les clients sont de plus en plus exigeants : ils veulent des applications finies, propres avec le moins de bogues possible. Mais je ne veux pas des « pisseurs de ligne » : chez 4D, l'équipe de programmeurs s'amuse ce qui n'empêche pas, lors de réunions pour la mise en place d'une nouvelle version, que les idées fusent de partout.

Que pensez-vous du mythe du développeur qui vit la nuit et ne mange que des pizzas ?

• Nous avons ici un ou deux programmeurs « décalés », arrivant à 11 heures du matin pour repartir tard le soir. Au cours de toutes ces années, j'en ai moins rencontré qu'on en parle. En fait, je pense que c'est surtout lié à un contexte professionnel comme la sortie d'un produit qu'il faut mener à terme.

Quid de la Net économie ?

• En fait, la micro informatique avait déjà changé de visage avant Internet. Il y a 10 ans, les utilisateurs étaient prêts

à supporter beaucoup de problèmes. Aujourd'hui, les applications sont devenues des outils de tous les jours. Lorsqu'il y a eu ce grand ram-dam en début d'année sur les dot.com, j'étais assez mécontent. Je suis peut-être de la « vieille école » mais j'aime la notion de conscience professionnelle, en d'autres termes, je veux vendre quelque chose qui existe. Le fait qu'il y ait eu une grande « claque » vérifie ce que je pensais. Rares sont les start-up qui vendent un véritable service.

Quels sont les programmeurs émergents ?

• Steve Jobs n'est pas vraiment un programmeur, mais il a une vision novatrice qui me plaît. Il essaie de créer quelque chose de nouveau.

Dans le même ordre d'idée, quels sont les grands applicatifs du siècle ?

• Des produits comme Visicalc de Dan Bricklin ont fait comprendre aux gens que le micro était un outil qui ne servait pas uniquement aux passionnés. Dans la même lignée, l'interface souris/fenêtre et tout ce qui a permis de faire évoluer les langages de programmation comme le concept « objet ».

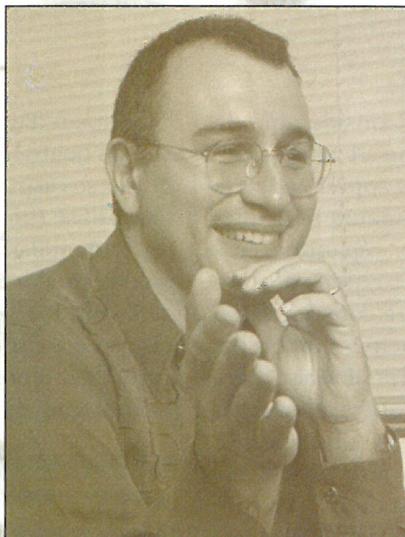
Quelles sont les recettes d'un bon logiciel ?

• Pour un utilisateur, un bon logiciel est un logiciel qui tourne avec le moins d'instructions possible, qui fait ce qu'on lui demande, et qui exige le minimum d'apprentissage. Pour l'éditeur, c'est celui qui se vend le mieux. On arrive souvent à concilier les deux, à trouver la bonne alchimie entre le beau et l'utile !

Quels sont selon vous les langages de programmation de demain ?

• Les nouveaux développeurs commencent de plus en plus

par des langages de haut niveau. Il y a vingt ans, on commençait par le Basic puis par l'assembleur pour comprendre la machine. Après, on a eu une étape avec des langages comme le Pascal et le C. Puis sont arrivés les L4G et les langages orientés objet comme le C++. Si le C++ perdure, je pense que les développeurs s'orienteront de plus en plus vers des L4G qui offrent désormais un niveau de contrôle équivalent à des langages moins faciles d'accès. Chez nous, pratiquement tout le monde travaille en C++ et en 4D. Un exemple : toute la gestion de l'interface de 4D est faite en 4D et tout le noyau est fait en C++.



➤ Laurent Ribardière, programmeur depuis l'âge de 15 ans.

➔ 4D et l'Internet Comme le passage de sa base de données sous Windows, Internet a grandement profité à l'entreprise de Laurent Ribardière. En effet, dès 1997, 4D a intégré la notion de serveur Web dynamique. « Notre but, avec 4D est de fournir un outil de développement d'applications, qu'elles soient Internet ou non. Nous avons poussé notre logique assez loin autour d'Internet puisque nous avons racheté la société StarNINE qui fait des serveurs web sous Mac. Son produit "WebSTAR" a été choisi par la Défense américaine pour sa fiabilité. »

Pour en savoir plus sur la société 4D et sa gamme de logiciels : www.4d.fr.

Que Forth soit avec toi

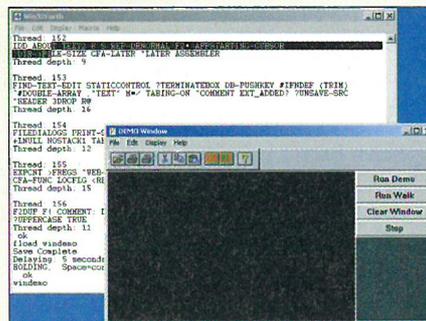
Jean-Marc Quéré
apiend@club-internet.fr

Face au nombre considérable de langages existants et à l'impossibilité de tous les appréhender, la majorité des développeurs ne s'intéressent qu'aux plus répandus. Les modes se succèdent au rythme des impératifs techniques et commerciaux. L'usage et les principes inhérents créant le savoir-faire, les réminiscences d'époques dites révolues apportent parfois des solutions inattendues, voire avant-gardistes aux problèmes quotidiens. L'informatique respire donc au fil des générations d'hommes et de machines. À titre d'exemple, l'évocation seule des dernières révolutions en matière d'Internet (Java, Perl, PHP) rappelle à nos mémoires l'emploi de langages de scripts et interprétés. Contemporains du glisser-déplacer et des environnements de développement à haut niveau d'intégration, partons à la recherche de concepts inhabituels. Volontaires polyglottes (et non troglodytes !), remontons les âges...

Aperçu

Il était une fois un... astronome, nommé Charles Moore, qui désirait un langage rapide, fiable et compact pour contrôler différents processus liés à son observatoire. Face au principe métaphysique, bien connu, selon lequel on n'est jamais mieux servi que par soi-même, il s'engagea dans le déve-

loppement d'un nouveau langage qualifié de quatrième (*Fourth*) génération de par ses aspects novateurs. L'ordinateur utilisé (un IBM 1130) n'acceptant pas d'identifiant de plus cinq lettres, la fin des années 60 vit donc l'apparition de Forth. Particulièrement bien organisé en matière de communication (en particulier d'observations), les milieux astronomiques l'adoptèrent rapide-



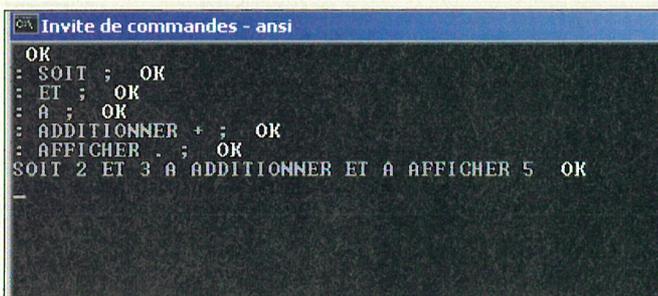
► Exemple d'application Windows en Forth.

ment. À tel point que sa diffusion s'effectuait commercialement dès le début des années 70 donnant naissance à la société Forth Inc. L'apparition de groupes d'utilisateurs animés par une passion ardente «*Forth Interest Group*» (www.forth.org) se manifesta par la multiplication des extensions et la mise en œuvre de procédure de normalisation. La branche française du FIG, nommée «*Jedi*», connut également ses heures de gloire. Après un (dernier ?) sursaut en 1991, époque du

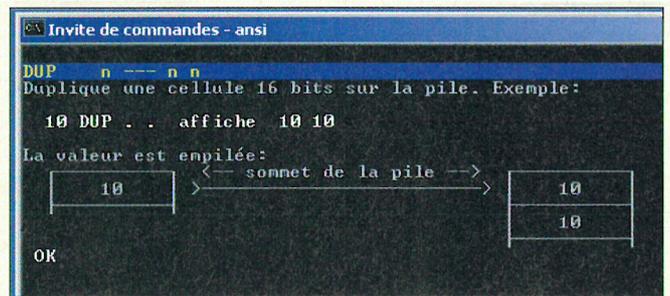
Turbo Pascal, Turbo C (Borland), etc., suite à la mise sur le marché du Turbo Forth (RemCorp), son utilisation se fit de plus en plus confidentielle. Actuellement, seuls quelques initiés l'exploitent (essentiellement en matière d'électronique – traitement de signaux – d'informatique embarquée et de robotique) pour ses qualités intrinsèques : grande vitesse et petite gourmandise en ressources (moins de 1Mo sur un PC sous DOS) ainsi que la faculté à générer des binaires à flasher sur des EPROM pour de multiples microcontrôleurs (*cross compilation*), etc.

En pratique

L'idée porteuse est liée à l'implémentation d'une machine virtuelle dotée d'une pile de traitement et de retour (dédiée à la gestion des branchements, des boucles,...) ainsi que des instructions permettant sa gestion. L'ensemble des commandes du langage sont basées sur cette architecture minimale. Le portage d'une plate-forme à une autre ne requiert que le portage de la machine virtuelle, soit principalement des commandes élémentaires liées à la pile : DUP (duplication de l'élément au sommet), SWAP (intervention avant-dernier élément avec le sommet), ROT (rotation cyclique à trois éléments), etc. L'ensemble des instructions peuvent ensuite être com-

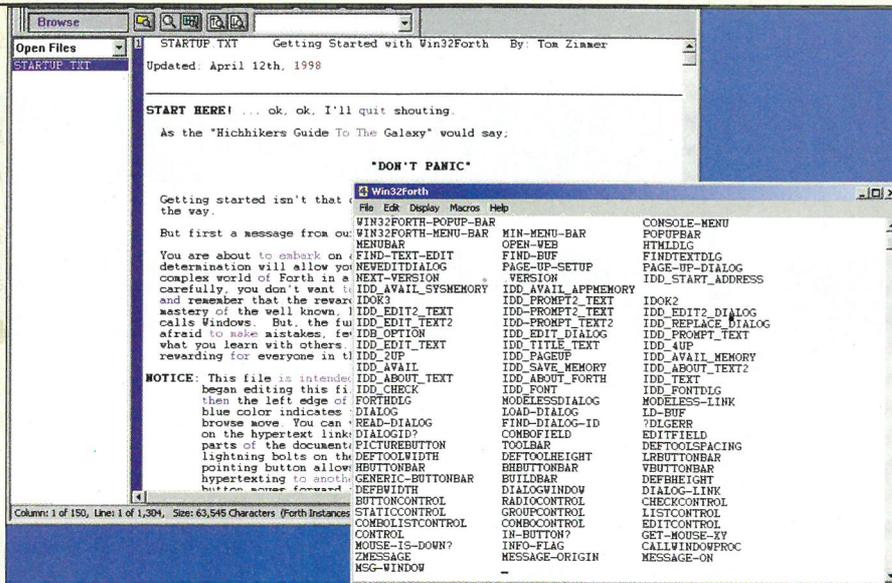


► En français dans le texte...



► Aide en ligne de Turbo Forth («*help dup*»).

pillées sur le dispositif cible pour reconstituer l'environnement de développement initial. Le langage utilise la notation postfixée (dite « polonaise inversée »). Ainsi l'opération « 1+2 », s'écrit « 1 2 + ». À la fois interpréteur et compilateur, il permet l'évaluation immédiate d'expressions en ligne de commande et la compilation de nouvelles définitions. Ajoutées au dictionnaire contenant l'ensemble des symboles (consultable par « WORDS »), elles sont liées à un contexte (« VOCABULARY »: vocabulaire). Plusieurs définitions d'un même symbole peuvent donc cohabiter dans des vocabulaires distincts. Une application peut ainsi être compilée à l'attention de contextes variés en optant pour le vocabulaire comportant l'implémentation des fonctions spécifiques à la cible considérée. Le corps du programme ne requiert aucune adaptation particulière (comme l'usage de l'écran du micro ou d'un terminal passif avec redéfinition des ordres d'entrées sorties). Les définitions, assimilables à des fonctions et ou procédures selon qu'elles retournent ou non un résultat sur la pile, peuvent être appelées individuellement pour être testées après adjonction des paramètres requis sur la pile. La mise au point des applications en est grandement facilitée car chaque entité peut être validée individuellement sans activer l'ensemble de l'application. Le niveau d'abstraction est laissé à l'appré-



> L'environnement de Win32Forth.

ciation du développeur. L'exemple suivant (scolaire) en est une parfaite illustration. Les mots « SOIT », « ET », « A » sont déclarés comme n'effectuant aucun traitement. Les mots « ADDITIONNER » et « AFFICHER » sont respectivement associés à l'addition « + » et à l'affichage d'une valeur numérique « . » (le point).

```
: SOIT ;
: ET ;
: A ;
: ADDITIONNER + ;
: AFFICHER ;
```

La séquence « SOIT 2 ET 3 A ADDITIONNER ET A AFFICHER » provoque l'affichage de 5. Le traitement effectivement effectué est « 2 3 + . ».

Révolutionnaire ?

Forth introduit de nombreuses notions novatrices comme la mise en œuvre d'une machine virtuelle pour limiter les contraintes liées à la rédaction du code, l'introduction de la gestion d'une pile qui s'avère redoutable, l'alias de commande (vecteur) modifiable en cours d'exécution, la possibilité d'intervenir sur le processus de compilation au sein des fonctions programmées pour modifier la syntaxe, la sémantique (compiler du pascal...), celle de créer de nouveaux types, d'adopter une orientation objet, ... Les versions les plus abouties, fournies avec leurs sources (Turbo Forth 83, par exemple), autorisent même la compilation de l'ensemble, en

totalité ou partiellement lors d'un déploiement. Suite à l'analyse de l'arborescence des définitions utilisées, leur extraction génère un nouveau noyau Forth restreint qui amorce automatiquement le symbole choisi sous la forme d'un programme autonome. À tout moment le noyau peut être sauvegardé en son état d'enrichissement pour conserver les apports effectués et constituer la base à un futur développement.

L'oubli de Forth a provoqué la chute des principales sociétés éditrices. Les produits – de qualité – sont donc passés dans le domaine public. ■

Apprendre

Une des plus accessibles et mieux documentées versions en français de Forth est TurboForth 83 de « feu » REMCORP (www.acbm.com/magasin.html).

Son principal développeur, Marc Petremann a intégré l'ensemble de la documentation en ligne. Il suffit d'entrer « help x » pour afficher les indications relatives à la commande « x ». Accompagné d'un manuel d'apprentissage, « Débutez en Forth » de Leo Brodie (Eyrolles), les premiers pas s'effectuent sans douleur en quelques heures. Adhérer à la démarche et vous ne tarderez pas à connecter des moteurs pas à pas sur le port parallèle de votre micro, voire des lecteurs de carte à puce ou tout autre type d'équipement. La compréhension de mécanisme lié au monde PC, requiert également une base technique solide. La consultation de « La Bible de l'IBM PC » de Michael Tischer (Micro Application) s'avère être un soutien indispensable. Preuve du dynamisme des adeptes du Forth, une version 32 bits pour Windows 9x/NT/2000 (avec documentation en anglais) peut être téléchargée à l'url suivant: [ftp://pweb.netcom.com/users/zforth](http://pweb.netcom.com/users/zforth).

Comment programmer... un (magnéto) ?

Vous n'avez pas d'enfant de 7 ans à la maison ? Dommage, parce que ç'eut été votre meilleur atout pour parvenir à enregistrer Matrix qui passe cette nuit à

2h35...

Carole Pitras

Deux constats à propos des magnétoscopes : lorsque vous en changez, tout est à réapprendre compte tenu de la très mauvaise volonté que mettent les fabricants (les marques les plus répandues sont Thomson, Sony, JVC, Panasonic et Sharp) à harmoniser le fonctionnement de leurs bestioles. Cependant, les traductions du japonais et du coréen dans les manuels d'utilisation se sont nettement améliorées en 15 ans. Le manuel étant souvent multilingue, il faut simplement trouver les pages en français.

L'enregistrement à l'heure H

Si vous n'avez pas envie de vous prendre la tête et que vous êtes noctambule ou insomniaque, la solution la plus simple consiste à attendre l'heure dite, d'insérer une cassette dans le magnéto (important la cassette, VHS dans la mesure du possible) et d'appuyer, sur le magnéto ou sa télécommande sur « record » ou « enregistrer », le bouton est en général facile à localiser car de couleur rouge. Ne pas oublier au préalable d'avoir mis le magnéto sur le bon canal sous

peine de vous retrouver avec les passionnantes émissions nocturnes « chasse et pêche » de TF1. Très utile pour vaincre les insomnies mais incontestablement décevant face à Matrix...

La programmation ponctuelle

Vous ne serez pas là lors du début du film, il faut donc vous résoudre à programmer à l'avance cet enregistrement. Première étape indispensable : que le magnéto soit à la bonne heure et à la bonne date (la date est essentielle pour les enregistrements la nuit). Certains magnétoscopes récents se mettent automatiquement à l'heure à partir de l'horloge d'une chaîne. Ensuite, allumez votre télé en mode magnéto afin de visualiser les données de la programmation sur l'écran. À partir du menu, opter pour programmation simple, hebdomadai-

re ou quotidienne. Mentionnez le canal, l'heure de début et de fin du programme (+10 minutes pour la fin, c'est toujours rageant de constater qu'on a bien le couloir de pub qui précède la diffusion mais pas toujours la fin), et vérifiez le jour, notamment pour les diffusions nocturnes : il est aisé de se faire piéger. N'oubliez pas de mettre une cassette et surtout, à la suite de la programmation

Les petites puces pullulent partout

il n'y a pas que les ordinateurs dans la vie. Le domicile regorge d'objets à puce qu'il n'est pas toujours aisé de faire fonctionner. Magnétoscopes, téléphones portables, fours à micro-ondes, machines à laver ..., ces engins se révèlent souvent obscurs pour le commun des mortels, développeur ou non. Mais, dorénavant, vous n'aurez plus aucune excuse car, tous les mois, nous détaillerons le fonctionnement de tous ces appareils censés nous faciliter l'existence. Merci qui ? Merci Programmez!

d'ÉTEINDRE le magnéscope. C'est là LE piège dans lequel tout le monde tombe un jour et ce n'est pas la peine d'accuser votre magnéscope de tous les maux, vous êtes l'unique coupable ET responsable.

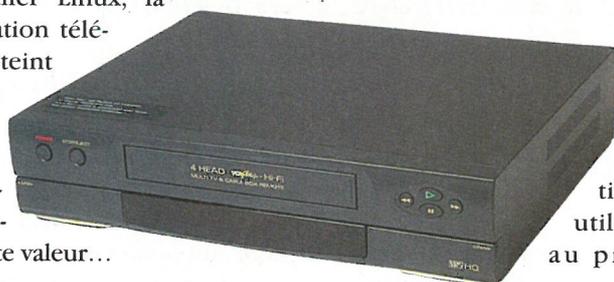
Si vous vous passionnez pour une émission ou une série dont la diffusion est hebdomadaire ou quotidienne, il est possible de programmer son magnéscope pour l'enregistrer régulièrement. Dans ce cas, n'oubliez pas de changer de cassette régulièrement.

Pour certains développeurs qui refusent d'utiliser une interface graphique pour installer Linux, la programmation téléviseur éteint reste une option SM qu'ils devraient pouvoir apprécier à sa juste valeur...

Enregistrer un film et en regarder un autre

Si vous n'avez pas programmé votre enregistrement, mettez votre magnéscope sur le bon canal, appuyez sur enregistrer puis prenez la télécommande de la télé, repassez sur le mode TV et zappez normalement. Si vous programmez l'enregistrement au préalable, le magnéscope ayant été éteint, vous pouvez zapper comme à votre habitude.

Le ShowView



Bizarrement, cette option pourtant très pratique est peu utilisée. Il faut au préalable se

munir d'un bon magazine TV (ce qui limite le choix). À chaque programme correspond un code de quatre à neuf chiffres. Mettez votre TV en mode magnéscope. Sur la télécommande du magnéscope appuyez sur ShowView, il va vous demander le code ; entrez-le puis suivez les instructions : seront alors affichés automatiquement (c'est magique) le canal, l'heure de début et de fin et le jour de diffusion. L'enregistrement se déclenche à l'heure annoncée par la chaîne qui diffuse le programme. Là-aussi, ajoutez à l'horaire de fin 5 à 10 minutes au moins, la diffusion commençant rarement à l'heure annoncée.

Et puisqu'il est difficile d'être exhaustif sur ce type d'engins en moins de 30 pages, le plus précieux conseil est d'avoir conservé le manuel d'utilisation à proximité de la TV, d'éteindre cette dernière et de songer à fabriquer un enfant de sept ans ... ■

NOVEMBRE 2000 N°5
UPGRADE magazine
LE MAGAZINE DU PC À LA CARTE
NOVEMBRE 2000 100% HARDWARE 29 F
PORTABLES
OSEZ L'UPGRADE !
OPTIMISEZ !
GRAVURE
Stocker ses films sur Video-CD
OVERCLOCKING
Processeurs FC-PGA : dépassez le Gigahertz
PLUS VITE !
Optimisez votre PC à moindres frais
SYSTÈME
Créez un PC multiboot
SOLUTIONS
Résoudre les conflits d'I/O et de DMA
LECTEURS DVD - Multi OS - Boîtiers moyennes tours - Cartes son - PC Portables
CARTES SON
5 modèles exceptionnels de 500 à 2 500F
COMPARATIF
LECTEURS DVD 12x-16
L'équipement idéal pour le cinéma
N°5

Nouveau !

Un rendez-vous à ne pas manquer

En vente actuellement chez votre marchand de journaux



Sortir du régime

pizza-cola

L'éditorial du mois dernier se concluait sur une interrogation passionnante : « ça mange quoi un programmeur ? ». Le principal critère déterminant du régime alimentaire du développeur est son âge. Plus il vieillit, plus son régime alimentaire se diversifie.

Carole Pitras

De la pizza/cole...

Adolescent, le développeur est un adepte de la *junk food* qui est rapide et peut se manger avec quatre doigts de la main gauche, tout en conservant la main droite sur le clavier. Manger n'est qu'une nécessité, et prendre un repas à table en compagnie d'autres humains est une insondable perte de temps lorsque tant de fascinantes lignes de code vous attendent. Le régime Pizza-Cola est alors un *must*, la pizza présentant l'avantage de laisser moins de miettes que le sandwich. Du point de vue diététique, c'est un désastre : des graisses animales, saturées qui plus est, et du sucre parfumé en guise de boisson. Pour limiter les dégâts, recommandons l'alternance entre quatre-fromages, quatre-saisons, calzone et margarita. Pour le soda, le sans-sucre est préférable, mais, objectivement, c'est beaucoup moins bon... Néanmoins, il semble bizarre que des développeurs, tellement sensibles à l'insupportable caractère hégémonique de Microsoft, soient si attachés à une marque de boisson qui ne l'est pas moins... ; Alors « *Think different* » and *Drink Pepsi* ou *Virgin*...

Heureusement, le jeune développeur limite l'absorption de nourriture au strict minimum, ce qui lui évite de devenir obèse. Ce type de comportement se retrouve jusqu'à la fin des études.

...au steak-frites café

À son entrée dans le monde du travail, le développeur continue la sociabilisation entamée durant ses études : il découvre alors une nouvelle boisson, qu'il alterne avec son soda adoré : le café, consommé là-aussi sans modération. Et lorsqu'il y a un restaurant à proximité, il commence à prendre ses repas de midi avec des congénères développeurs autour d'une table, et à varier son régime alimentaire : croque-monsieur, steak-frites, pâtes à la bolognaise ou à la carbonara voire pour les plus audacieux - mais c'est rare - poisson/riz. D'un point de vue diététique ça reste moyen : encore trop de graisse, et notamment de graisses animales, mais la découverte des glucides sous la forme de féculents (pâtes, riz, pomme de terre) est encourageante. Le soir, en revanche, le couple pizza/cole reste incontournable, soit surgelé soit livré à domicile.

Les solutions alternatives

Pourtant, d'autres aliments peuvent être livrés à domicile. Le site eatonline.fr est une mine : il répertorie les possibilités de livraison à proximité du bureau ou du domicile. On trouve de la cuisine traditionnelle française, du

chinois, du japonais, de l'indien, du libanais, du tex-mex, du thaï, un couscous, un plateau de fruit de mer, une paella, un poulet rôti, des sandwiches plus évolués que le Paris-beurre-double-face et des salades. En alternant ces différentes nourritures, on peut parvenir à un régime alimentaire équilibré : le couscous est particulièrement complet avec féculents, légumes et viande (glucides et protides), la cuisine vapeur chinoise permet de compenser les excès de graisse de même que les sushis ou sashimis japonais qui apportent plein de bonnes choses à votre petit organisme malmené.

Pour les adeptes du surgelé, on trouve dans les rayons spécialisés des supermarchés des légumes excellents et des plats complets, comme des poissons en papillotes, qui présentent l'avantage de se préparer facilement au micro-ondes. Et désormais entre Telemarket, Ooshop et autres, plus besoin de se déplacer, on peut commander en ligne.

À partir de 30/35 ans, souvent sous l'influence discrète mais persistante d'un membre courageux de la gent féminine, le développeur retrouve un régime omnivore correspondant à celui de ses congénères non développeurs, à savoir qui inclut des éléments aussi exotiques que des fruits et légumes. L'habitude du Coca est plus pernicieuse, mais l'âge aidant la découverte de certains vins n'est pas à exclure. ■



Recette de Noël : la Dinde au Whisky*

- Acheter une dinde d'environ 5 kg pour 6 personnes et une bouteille de whisky, du sel, du poivre, de l'huile d'olive, des bardes de lard.
- La barder de lard, la ficeler, la saler, la poivrer et ajouter un filet d'huile d'olive.
- Faire préchauffer le four thermostat 7 pendant dix minutes.
- Se verser un verre de whisky pendant ce temps-là.
- Mettre la dinde au four dans un plat de cuisson.
- Se verser ensuite 2 verres de whisky et les boire.
- Mettre le thermostat à 8 après 20 minutes pour la saisir.
- Se bercer 3 berres de whisky.
- Après une demi beurre, fourrer l'ouvrir et surveiller la cuisson de la pinde.
- Brendre la bouteille de biscuit et s'enfiler une bonne rasade derrière la bravate - non - la cravate.
- Après une demi heure de blus, tituber jusqu'au bour. Ouvrir la putain de borte du bour et reburner - non - revourner - non - recourner - non - enfin, mettre la guinde dans l'autre sens.
- Se prûler la main avec la putain de borte du bour en la refermant - bordel de merde.
- Essayer de s'asseoir sur une putain de chaise et se re-
- verdir 5 ou six whisky de verres ou le gontraire, je sais blus.
- Buire - non - luire - non - cuire - non - ah ben si - cuire la bringue bandant 4 heures.
- Et hop, 5 berres de plus. Ca fait du bien par où que ça passe.
- R'tirer le four de la dinde.
- Se rebercer une bonne goulée de whisky.
- Essayer de sortir le bour de la saloperie de pinde de nouveau parce que ça a raté la première fois.
- Rabasser la dinde qui est tombée bar terre et bar inadverantce. L'ettuyer avec une saleté de chiffon et la foutre sur un blat, ou sur un clat, ou sur une assiette. Enfin, on s'en fout...
- Se ruiner la façade à cause du gras sur le barrelage, ou le carrelage, de la buisine et essayer de se relever.
- Décider que l'on est aussi bien par terre et binir la mouteille de rhiskey.
- Ramper jusqu'au lit, dorbir toute la nuit.
- Manger la dinde froide avec une bonne mayonnaise, le lendemain matin et nettoyer le bordel que tu as mis dans la cuisine la veille, pendant le reste de la journée.

Burp !

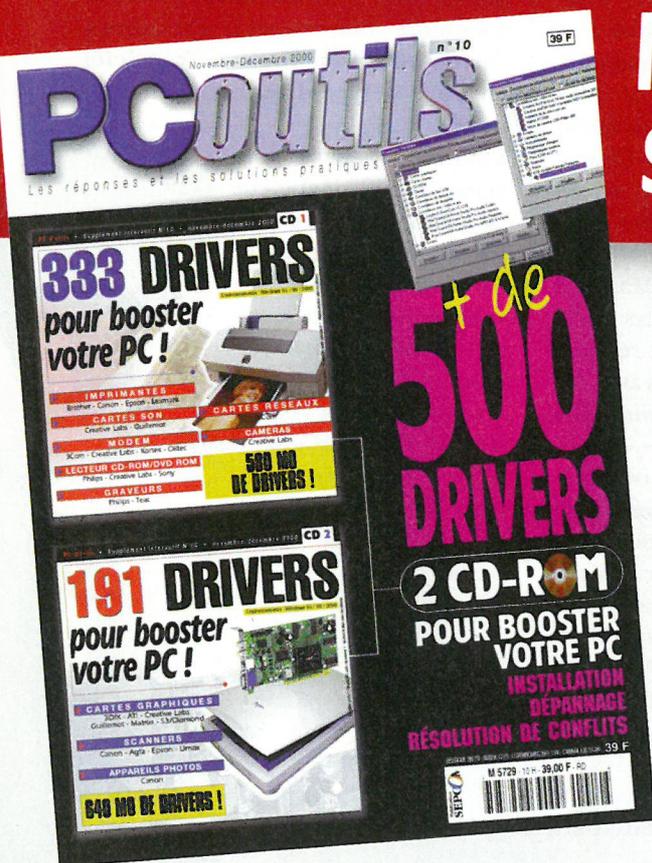
*A consommer avec modération

Pilotez en toute sécurité !



Plus de 500 drivers pour votre PC et vos périphériques : 1 220 Mo de logiciels pour environnements Windows 95, 98 et 2000 !

- ✓ Imprimantes
- ✓ Cartes son
- ✓ Modem
- ✓ Lecteur CD-ROM, DVD
- ✓ Graveurs
- ✓ Cartes graphiques
- ✓ Scanners...



ACTUELLEMENT EN VENTE CHEZ VOTRE MARCHAND DE JOURNAUX

Avoir le nez fin

Face à la pénurie de développeurs, les sociétés informatiques, et en premier lieu les SSII, rivalisent d'ingéniosité pour recruter. Petit tour d'horizon des pratiques actuelles, des plus loufoques aux plus sérieuses.

Carole Pitras

carole.pitras@programmez.com

Des innovations parfois étranges peuvent se décliner façon Cyrano de Bergerac.

Gourmand ! SII, société de service spécialisée dans l'informatique industrielle, distribue dans les cafés aux alentours des écoles d'ingénieurs ou dans ceux à proximité de ses concurrents des sucres dont l'emballage indique *SII recrute* et les profils recherchés.

Snob ! Pégase avait adopté en mars dernier une nouvelle approche pour inciter les dépôts de CV sur son site. Chaque mois, parmi les CV remis, qui doivent bien sûr avoir un rapport avec l'activité de l'entreprise, est tiré au sort, par huissier, un candidat qui gagne un week-end pour deux dans un « Relais & Châteaux »

Élégant ! Chez Akazi Technologies, le personnel, de la standardiste au président, se baladent et vont déjeuner dans les restaurants d'entreprises de la zone industrielle où la société est implantée avec un T-shirt « Akazi recrute/www.akazi.com ». Il paraît que les CV affluent : ils racontent « *L'intérêt éveillé dans la région, sur un marché du travail pourtant fortement concurrentiel, a même entraîné un candidat*

pour recruter

à faire le siège des bureaux de la société pour y déposer son CV dès le lendemain de la première action. »

Ludique ! Univers technologie avait organisé sur l'année universitaire 99/2000, plutôt à l'attention des étudiants donc, un jeu interactif, *Luna et les sphères magiques* qui se déroulait en sept étapes. Les meilleurs gagnaient des cadeaux... et une proposition d'embauche.

Sportif ! Il existe des rencontres entre clubs de foot d'entreprises. Groupe Actif passe donc des annonces pour recruter des informaticiens dans le quotidien sportif *l'Équipe*.

– **Festif !** Afin de recruter pour son activité multimédia plus de 250 ingénieurs dans la région Nord, Atos a récemment organisé une « rencontre recrutement » dans un café de Lille. L'invitation mentionnait « *Open bar et ambiance garantis !* ». Pour le boulot, par la suite, l'ambiance n'est en revanche pas garantie et l'« open bar » sûrement pas encouragé...

Communiquant ! CSC Peat Marwick a en août et septembre derniers multiplié la communication sur le recrutement : affichage dans les gares avec des slogans comme « *Départ immédiat pour la nouvelle économie* » « *Destination nouvelle économie* » ou encore « *Nouvelle économie, nouveau moi* ». Ces affiches annonçaient une journée de rencontres autour de l'emploi et était relayée par une campagne radio sur BFM (est-ce vraiment la radio écoutée par les jeunes diplômés ?). CSC n'est pas la seule société à avoir lancé des campagnes d'affichage pour recruter : Brime technologies et Groupe OCEI se sont largement affichés dans les couloirs du métro parisien.

Dans la rubrique un peu plus sérieuse, les partenariats avec les filières de for-

mation se multiplient, ainsi que le recours aux ingénieurs étrangers.

Fi system a mis en place une formation avec l'École des Mines d'Alès. Les 10 ingénieurs qui la suivent sur un semestre intègrent Fi system et s'engagent à rester dans l'entreprise pendant deux ans.

Groupe Open a signé un partenariat avec l'Université de Moscou : des candidats, bilingues, sont formés durant trois à six mois aux outils de développement web avant de rejoindre les équipes d'**Innetis**, la filiale service du groupe.

Certaines sociétés ouvrent des filiales dans des pays francophones pour profiter de ressources humaines qualifiées à moindre coût comme la web agency **NovosysCom** en Tunisie. **Transiciel** envisage ce type d'approche en Roumanie.

D'autres sociétés font venir des informaticiens de l'étranger : **Sys-com** a mis en place une filiale de formation pour des ingénieurs marocains, qui pourront par la suite venir travailler en France. **Aston** a recruté des ingénieurs malgaches et **Cadextan** embauche également de nombreuses personnes de nationalités différentes. ■

