

Developpez

Magazine

Edition de Septembre Octobre 2006.

Numéro 6.

Magazine en ligne gratuit.

Diffusion de copies conformes à l'original autorisée.

Réalisation : Cédric Chatelain

Rédaction : la rédaction de Developpez

Contact : magazine@redaction-developpez.com

Index

Linux	Page 2
Java	Page 7
Web & dev.web	Page 13
DotNet	Page 16
C & C++	Page 20
XML	Page 25
SGBD	Page 28
Windows	Page 31
Visual Basic	Page 35
Liens	Page 41

Editorial

La rentrée est passée et le magazine de Developpez.com revient avec son 6ème numéro.

N'hésitez pas à passer sur les forums pour donner votre avis ou faire des suggestions pour cette publication

Tutoriel JAVA

Les strings se déchaînent



L'utilisation de l'objet String dans son code source est extrêmement aisée mais hélas beaucoup trop souvent source de contre-performances. L'obtention d'un bytecode efficace ne sera possible que si vous apprenez à maîtriser les Strings. Afin d'y parvenir, nous allons tout d'abord voir comment ces objets sont gérés par les compilateurs.

par **Romain Guy**

Page 7

Article Windows

Firefox et ses meilleures extensions



Grâce à un développement continu (actuellement en version 1.5.0.6) et à l'ajout d'une infinité de fonctionnalités, Firefox continue de séduire les gens et gagne peu à peu des "parts de marché".

par **Louis-Guillaume Morand**

Page 31



Les derniers tutoriels et articles

Installer MySQL 5.0.24 sur FreeBSD

1. Avant-Propos

L'utilisation de MySQL en tant que système de bases de données est un bon choix étant donné les principaux avantages de ce dernier. En effet, MySQL est un logiciel gratuit, open-source, rapide, robuste et largement supporté par une communauté de développement sans cesse grandissante. Sa grande utilisation sur Internet n'est plus à démontrer, et une fois couplé à PHP, la puissance qui en résulte suffit à la réalisation de la plupart des projets de chacun.

Au cours de cet article, je vous expliquerai pas à pas comment installer la dernière version stable de MySQL (actuellement la 5.0.24) sur un système BSD. Tout ce dont vous avez besoin est un compte root et une connexion à Internet si vous envisagez de télécharger les sources de MySQL sur la toile.

2. Téléchargement des sources

La première chose à faire est donc de télécharger les sources de MySQL. Vous les trouverez sur le site officiel du logiciel, dans la section *Downloads* de la *Developer Zone* (<http://dev.mysql.com/downloads/>). Cliquez sur le lien nommé *Current Release*. Les sources se situent tout en bas de la page, sous le nom de *Source downloads*. Étant donné que nous utilisons FreeBSD, le plus simple pour nous sera de prendre l'archive tarball (.tar.gz). La taille de l'archive est de 19 Mo.

Au cours de mon explication, je supposerai que les sources se trouvent dans le répertoire `/usr/src`.

3. Création de l'utilisateur mysql

Avant de compiler MySQL à proprement parler, il est nécessaire de créer le compte utilisateur mysql, identité sous laquelle le serveur MySQL sera lancé. L'utilisateur mysql appartiendra au groupe nommé lui aussi mysql que nous créerons également.

Création de l'utilisateur mysql

```
# pw groupadd mysql
# adduser
Username: mysql
Full name: mysql
Uid (Leave empty for default):
Login group [mysql]: mysql
Login group is mysql. Invite test into other groups? []:
Login class [default]:
Shell (sh csh tcsh nologin) [sh]:
Home directory [/home/mysql]: /usr/local/mysql/
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
```

```
Lock out the account after creation? [no]:
Username : mysql
Password : *****
Full Name : mysql
Uid : 1004
Class :
Groups : mysql
Home : /usr/local/mysql/
Shell : /bin/sh
Locked : no
OK? (yes/no): yes
adduser: INFO: Successfully added (mysql) to the user
database.
Add another user? (yes/no): no
Goodbye!
#
```

Les valeurs entre crochets ([yes], [sh], ...) sont les valeurs par défaut. Cela signifie que si vous n'entrez aucune valeur et que vous validez, c'est cette valeur qui sera utilisée. Notez que le répertoire utilisateur de mysql est en fait le répertoire où l'on installera MySQL par la suite (`/usr/local/mysql`).

Dernier petit détail avant la compilation : il faut créer 3 sous-dossiers dans `/usr/local/mysql`.

```
# mkdir /usr/local/mysql/data
# mkdir /usr/local/mysql/tmp
# mkdir /usr/local/mysql/var
# chmod 775 /usr/local/mysql/data /usr/local/mysql/tmp
/usr/local/mysql/var
```

4. Compilation et installation de MySQL

Nous allons maintenant décompresser l'archive que nous avons téléchargée et nous rendre dans le dossier créé :

Décompression

```
# cd /usr/src
# tar -xzf mysql-5.0.24.tar.gz
# rm mysql-5.0.24.tar.gz
# cd mysql-5.0.24
```

On lance le script *configure* en spécifiant le répertoire où l'on souhaite installer MySQL ainsi que celui où seront stockées les données (*localstatedir*). On indique aussi l'emplacement du socket qui sera créé par le serveur MySQL. Pour voir toutes les options possibles, effectuez `./configure --help`.

Script configure

```
# ./configure --prefix=/usr/local/mysql --
localstatedir=/usr/local/mysql/data --with-unix-socket-
path=/usr/local/mysql/tmp/mysql.socket
```

On finit par compiler et installer.

Compilation et installation

```
# make
# make install
```

5. Quelques vérifications s'imposent...

Maintenant que MySQL est installé, nous allons pouvoir le lancer. Cela nécessite d'installer les bases de données préexistantes en effectuant la commande suivante :

```
# cd /usr/local/mysql/bin
# ./mysql_install_db --user=mysql
```

On peut donc enfin lancer le serveur :

Lancement du serveur MySQL

```
# ./mysqld_safe --user=mysql &
```

MySQL inclut quelques outils utiles qui vont nous permettre de vérifier que notre serveur fonctionne bel et bien :

```
# ./mysqladmin version
Server version 5.0.24
Protocol version 10
Connection Localhost via UNIX socket
UNIX socket /usr/local/mysql/tmp/mysql.socket
Uptime: 1 min 25 sec
# ./mysqlshow
+-----+
| Databases |
+-----+
| information_schema |
| mysql |
| test |
+-----+
#
```

Pour stopper le serveur MySQL, effectuez la commande suivante :

Arrêt du serveur MySQL

```
# ./mysqladmin -u root shutdown
```

6. Remplacement du mot de passe root

Le compte root créé à l'installation du serveur MySQL possède tous les droits. Il peut aussi bien supprimer n'importe quelle base de données qu'ajouter ou supprimer un utilisateur. Cela implique la mise en place d'un mot de passe pour ce compte qu'il ne faut pas laisser entre toutes les mains.

Le changement de mot de passe se fait via *mysqladmin*. Lorsqu'un mot de passe est ajouté, il est crypté et est ensuite ajouté dans la table *user* située dans la base de donnée nommée *mysql*.

Changement du mot de passe root

```
# ./mysqladmin -u root password monpass
```

Ainsi, le nouveau mot de passe du compte root sera *monpass*.

Enfin, n'oubliez pas que l'ajout d'un mot de passe implique l'ajout du flag **-p** dans chacune des commandes relatives à MySQL. Par exemple :

```
# ./mysql -u root -p
# ./mysqladmin -u root -p shutdown # ./mysqladmin -p
version
```

Si vous oubliez ce dernier, le message d'erreur suivant apparaîtra :

```
error: 'Access denied for user 'root'@'localhost' (using
password: NO)'
```

7. Configuration de MySQL à l'aide du fichier *my.cnf*

Je vais terminer par l'utilisation du fichier *my.cnf*. L'utilisation de ce fichier est facultative mais elle permet de préciser quelques options comme le port d'écoute du serveur MySQL.

Le fichier original se trouve dans le répertoire */share/mysql* de MySQL. Dans notre cas, il se situe donc dans */usr/local/mysql/share/mysql*. En parcourant le dossier, vous remarquerez que le fichier *my.cnf* en lui-même n'existe pas. En effet, nous retrouvons à la place quatre fichiers : *my-small.cnf*, *my-medium.cnf*, *my-large.cnf* et enfin *my-huge.cnf*. Vous allez devoir choisir un de ces quatre fichiers d'après la puissance de votre machine pour ensuite le placer dans */etc* sous le nom de *my.cnf*. Ma machine possédant 320 Mo de RAM, je choisirai le fichier *my-medium.cnf*.

```
# cd /usr/local/mysql/share/mysql
# cp my-medium.cnf /etc/my.cnf
```

Le fichier *my.cnf* sera pris en compte lors de la prochaine exécution du serveur. Vous pourrez voir que le fichier est divisé en deux parties : **[client]** pour le client et **[mysqld]** pour le serveur.

La partie client ne possède pas beaucoup d'options, quoique vous puissiez en rajouter. Son principal intérêt est de se connecter rapidement à un serveur MySQL défini dans le fichier *my.cnf*. Dans l'exemple qui suit, le client *mysql* se connectera à localhost sur le port 3316 en utilisant le compte *paul* créé auparavant.

my.cnf

```
[client]
host = localhost
port = 3316
user = paul
password = sonpass
socket = /usr/local/mysql/tmp/mysql.socket
```

Nous avons donc rajouté deux champs ; *host* qui spécifie l'hôte auquel le client MySQL se connecte, et *user* qui définit l'utilisateur utilisé par défaut par le client.

Coté serveur, on peut également ajouter le champ *user*. Ainsi, nous n'aurons plus besoin de le spécifier lorsque nous lancerons *mysqld_safe*.

Je vais m'arrêter ici concernant l'utilisation de *my.cnf*. Vous trouverez d'autres informations à ce sujet sur le site officiel de MySQL (<http://www.mysql.com>).

L'installation de MySQL est maintenant achevée. N'hésitez pas à consulter la documentation si vous avez des questions, elle répondra à la majorité d'entre elles. Cependant, s'il reste des questions auxquelles vous ne trouvez aucune solution, n'hésitez pas à me contacter ou à vous rendre sur le forum de [developpez.com](http://www.developpez.com) (<http://www.developpez.net>) qui possède une section SQL.

Retrouvez la suite de l'article de Frédéric Mereu : [Lien1](#)

Installation des utilitaires HP sur Debian GNU/Linux

Ce document a pour but de vous aider à mettre en place les agents HP sur la plateforme Debian. Nous verrons comment, grâce à diverses manipulations, installer et configurer les agents en vue de l'intégration à Insight Manager.

HP ne publiant pas encore des versions pour Debian de ces agents, j'ai bricolé un petit peu pour réussir à les faire fonctionner. Ces agents permettent la remontée des informations sur une console (Insight Manager), et permet, entre autre, d'être averti dès qu'un incident se produit.

Je mets ce document à disposition en espérant qu'il vous sera utile, et qu'il permettra toutefois d'économiser votre temps en recherche d'informations.

Ce document est basé sur diverses informations trouvées sur internet, ainsi que sur une mise en pratique de cette installation.

> Matériel utilisé

Serveur HP Proliant DL360 G4

Contrôleur de disque SATA non-RAID (RAID-SATA pris en charge par le système d'exploitation)

Systeme Ubuntu Linux 5.04 (basé sur Debian 3.1)

Kernel 2.6.10

> Versions utilisées

HP Management Agents (hpsasm) : version 7.3.0b

HP Management Homepage (hpsmh) : version 2.1.0

1. Pré-requis

alien

alien est l'outil disponible sur Debian qui permet la conversion des rpm en package Debian (.deb). Nous en aurons besoin par la suite pour convertir les rpm fournis par HP.

```
$ apt-get install alien
```

snmp

snmp est l'outil qui permet d'obtenir des informations sur le serveur depuis des machines distantes. Il sera utile notamment pour l'interrogation du serveur par Insight Manager.

```
$ apt-get install snmpd
```

tcl

tcl est requis par hpimlview.

```
$ apt-get install tk8.3
```

2. HP System Management Agents

2.1. Récupération et conversion

Récupération du package chez hp

```
ftp://ftp.compaq.com/pub/products/servers/supportsoftwar  
e/linux/  
ou  
wget  
ftp://ftp.compaq.com/pub/products/servers/supportsoftwar  
e/linux/hpsasm-  
7.3.0b-63.ul10.i386.rpm
```

Remarque : assurez-vous de prendre la version pour United Linux. La version utilisée dans ce tutoriel est la version 7.3.0b.

Conversion du rpm avec alien

```
$ alien -t -c hpsasm-7.3.0b-63.ul10.i386.rpm  
$ tar -xzf hpsasm-7.3.0.tgz
```

Les fichiers sont maintenant décompressés dans le sous répertoire *hpsasm- 7.3.0*. A présent, nous devons effectuer quelques modifications afin de faire fonctionner les scripts d'installation et d'initialisation sur plateforme Debian.

2.2. Installation

Adaptation à la plateforme Debian

Copions les fichiers à leur emplacement de destination.

```
$ cp etc/init.d/hpsasm /etc/init.d/  
$ cp -R opt/* /opt/  
$ cp sbin/* /sbin/  
$ cp -R usr/share/* /usr/share/  
$ ln -s /bin/pidof /sbin/pidof
```

Remarque : hpsasm recherche *pidof* dans /sbin, si il n'est pas présent il suffit de créer le lien symbolique.

Modifions ces fichiers pour les rendre compatibles à la plateforme.

- etc/init.d/hpsasm : supprimez (ou commentez) les lignes 19,28,47,48,52,53,57,58 (cf Annexe pour les lignes commentées)
- install/predoinst.sh : modifiez la ligne 48 comme suit

```
useradd -u 79 -g 79 -s /sbin/nologin -M -r -d  
/opt/hp/hpsmh hpsmh >  
/dev/null 2> /dev/null  
par  
useradd -u 79 -g 79 -s /sbin/nologin -M -d /opt/hp/hpsmh  
hpsmh >  
/dev/null 2> /dev/null
```

Remarque : L'option -r de la commande useradd n'est pas reconnue sous Debian.

Installation

Nous sommes maintenant prêts à exécuter le script de préinstallation. Allons-y !

```
$ install/predoinst.sh
```

Si tout se passe bien, il ne doit pas y avoir de sortie à l'écran. Il se peut en revanche que le script crie si des versions antérieures sont déjà installées. Il faut dans ce cas procéder à une désinstallation avant de pouvoir continuer. Une fois les désinstallations effectuées, vous pouvez relancer le script jusqu'à ce qu'il n'y ait plus d'erreur.

Pour terminer l'installation il suffit d'exécuter le script d'installation.

```
$ install/doinst.sh
```

2.3. Configuration

On tient le bon bout ! Activons le service dans un premier temps.

```
$ /etc/init.d/hpsm activate
```

Remarque : cette phase semble être déjà faite par le script doinst.sh. Attendez-vous à obtenir ce type de résultat, qui diffère selon le matériel de votre serveur..

```
NOTE: New snmpd.conf entries were added to the top of
/etc/snmp/snmpd.conf
```

```
NOTE: New cma.conf entries were added to the top of
/opt/compaq/cma.conf
```

Following environment variables were used to configure snmpd/agents:

```
CMAAGENTEXCLUDESTR
CMALOCALHOSTRWCOMMSTR
CMALOCALHOSTROCOMMSTR
CMAMGMTSTATIONIPORDNS
CMAMGMTSTATIONRWCOMMSTR
CMAMGMTSTATIONROIORDNS
CMAMGMTSTATIONROCOMMSTR
CMADEFTRAPCOMMSTR
CMASYSCONTACT
CMASYSLOCATION
```

```
Starting HP Server Management Drivers and Agents, please
wait ...
```

Passons à la configuration proprement dite.

```
$ /etc/init.d/hpsm configure
```

```
Welcome to the hp System Health Application and Insight
Management Agent(hpsm) package installation. This
package contains the hp Advanced Server Management
Application(hpsamd) and hp's SNMP agents. This package
is intended to only function on hp servers with either
the ProLiant ASM (0x0E11A0F2) ASIC or the ProLiant iLO
Advanced Server
```

```
Management (0x0E11B203) ASIC.
```

```
Do you wish to continue? <y/n> (blank is y)
```

Go !

The startup scripts in this package also control the loading of modules in this and other hp management packages (i.e. hprsm). Some of those management modules are distributed under a non-GPL license. When these non-GPLed modules are loaded the kernel's tainted flag (see /proc/sys/kernel/tainted) may be set. If you choose not

to load them, some management functionality will be lost.

Do you want to load the hp modules even though they may "taint"

```
your kernel? <y/n> (Blank is y)
```

Cette question est importante si vous comptez installer d'autres modules non- GPL plus tard. Personnellement je n'ai pas installé d'autres agents que ceux fournis par HP, j'ai conservé la réponse par défaut.

Some agents are configured not to start. The startup rules are set up in cma.conf. The following agents are excluded from running: cpqasm cmhp cmhp cmhp

```
Do you wish to change this (y/n) (Blank is n):
```

Le package contient tous les agents pour les serveurs HP, mais vous n'avez pas forcément besoin de tous. Par exemple j'ai désactivé l'agent qui vérifie le contrôleur SCSI puisque mon serveur possède un contrôleur SATA. C'est à vous de voir ce dont vous avez besoin.

Remarque : je vous conseille ici de laisser la réponse par défaut, nous modifierons cela après le premier lancement.

You will now be asked a series of SNMP configuration questions. How you answer these question will affect the way SNMP behaves. Configuring SNMP could have security implications on your system. If you are not sure how to answer a question, you can abort by pressing <Ctrl-c> and no changes will be made to your SNMP configuration.

Enter the localhost SNMP Read/Write community string (one word, required, no default): Enter localhost SNMP Read Only community string

(one word, Blank to skip):

Enter Read/Write Authorized Management Station IP or DNS name

(Blank to skip):

Enter Read Only Authorized Management Station IP or DNS name (Blank to skip):

Enter default SNMP trap community string (One word; Blank to skip):

Enter SNMP trap destination IP or DNS name (One word; Blank to skip):

The system contact is set to syscontact Root <root@localhost> (configure /etc/snmp/snmpd.local.conf)

Do you wish to change it (y/n) (Blank is n):

The system location is set to syslocation Unknown (configure /etc/snmp/snmpd.local.conf)

Do you wish to change it (y/n) (Blank is n):

```
NOTE: New snmpd.conf entries were added to the top of
/etc/snmp/snmpd.conf
```

```
NOTE: New cma.conf entries were added to the top of
/opt/compaq/cma.conf
```

```
Starting HP Server Management Drivers and Agents, please
wait ...
```

Ici il faut faire attention à ce qu'on fait. Je ne connais pas bien le fonctionnement de snmp (au niveau sécurité) à l'heure actuelle, je vous conseille de vous renseigner sur les risques de sécurité liés à ces paramètres. Ils seront utiles lorsque vous désirerez ajouter votre serveur dans Insight Manager (afin qu'il puisse récupérer les informations sur le matériel).

localhost SNMP Read/Write community : private

localhost SNMP Read community : public

Read/Write Authorized Management Station IP :

192.168.0.10 (poste avec Insight Manager)

Read Only Authorized Management Station IP :

192.168.0.10 (poste avec

Insight Manager)

SNMP trap : (vide)

SNMP trap destination : (vide)

System contact : adresse email de l'administrateur

Voilà la configuration est écrite dans les différents fichiers et l'installation est terminée.

2.4 Premier lancement

Lors du premier lancement nous allons regarder que tout se passe bien. Mais avant ceci ajoutons le script de démarrage au boot de la machine.

```
$ update-rc.d hpsasm defaults
```

Ceci va activer le démarrage des agents au lancement du serveur et les désactiver lors de son reboot ou de son arrêt. Maintenant nous pouvons lancer les agents (à priori ils doivent déjà être

lancés puisque l'installation s'est bien déroulée, mais mieux vaut s'en assurer).

```
$ /etc/init.d/hpsasm start
```

Les fichiers de log se trouvent à l'emplacement `/var/spool/compaq`. Regardez les fichiers **hpsasm.log** (qui ne devrait pas présenter d'erreur, il log le lancement du daemon) et **cma.log** (qui présentera les erreurs retournées par les agents).

Par exemple l'agent `cmascsid` produira des erreurs sur les machines possédant un contrôleur SATA (et donc aucun contrôleur SCSI). Dans ce cas il suffit de désactiver le démarrage de l'agent dans le fichier `/opt/compaq/cma.conf`.

```
exclude cmhp cmascsid # (exemple)
```

Une fois que toutes les erreurs seront résolues, les agents fonctionneront correctement.

Remarque : les agents utilisent snmp, il faut prêter attention à l'ordre de démarrage. Le daemon `snmpd` doit démarrer avant les agents, par exemple :

```
S20snmpd S21hpsasm
```

Retrouvez la suite de l'article de David Burgermeister : [Lien2](#)

Vu sur les Blogs

[Construction d'une FAQ Aix ! \(Lien3\)](#)

La FAQ Aix, version alpha bêta epsilon (rayez les mentions inutiles) est sortie dans sa première monture.

J'ai laissé exprès les questions en cours (j'ai les données dans la tête, faut juste que je pose tout ça au propre) pour voir la structure globale.

Toute aide est acceptée même pour modifier ce qui est fait ou rajouter d'autres parties comme la manipulation de l'ODM,

[Comment se connecter en ssh sans mot de passe ?](#)

Le but de l'opération est que l'utilisateur A de la machine Source se connecte comme utilisateur B de la machine Destination. L'opération est un transfert de clé publique entre votre Source et votre Destination.

Sur la Source

En tant qu'utilisateur A, vous devez générer une paire de clé RSA privée/publique :

```
ssh-keygen -t rsa -b 'nombre de bits de la clé' -N "";
```

Je conseillerai 4096 pour la complexité de la clé.

Deux fichiers sont apparus dans `~/.ssh` :

`id_rsa`

sauvegarde sous `mksysb & Cie`.

N'hésitez surtout pas à venir sur ce [post](#) nous aider !

Katyucha

Retrouvez le billet de Sylvain Luce : [Lien4](#)

`id_rsa.pub`

`id_rsa` correspond à la clé privée

`id_rsa.pub` correspond à l'autre clé, celle publique.

Sur la Destination

La clé publique va être recopiée sur la Destination dans `~B/.ssh/authorized_keys` :

Vous avez un fichier dans le format suivant

```
ssh-rsa 'la clé publique' A@Source_
```

Maintenant depuis la machine Source en utilisateur A, il vous suffit de faire :

```
ssh B@Destination
```

Et voilà !

Retrouvez le billet de Sylvain Luce : [Lien5](#)





Les derniers tutoriels et articles

Les Strings se déchaînent

Les Strings sont un des objets les plus employés de Java. A tel point qu'il est même impossible de créer une application Java sans en faire usage. Voyons comment les utiliser de manière optimale...

1. Introduction

L'utilisation de l'objet String dans son code source est extrêmement aisée mais hélas beaucoup trop souvent source de contre-performances. L'obtention d'un bytecode efficace ne sera possible que si vous apprenez à maîtriser les Strings. Afin d'y parvenir, nous allons tout d'abord voir comment ces objets sont gérés par les compilateurs.

2. 1 + 1 = 2

Les compilateurs Java permettent d'additionner directement les Strings entre elles, ce qui va à l'encontre même de leur nature. Les Strings sont en effet des objets et Java ne permet pas de surcharger les opérateurs. Que se passe-t-il donc ? En réalité, tout dépend du compilateur que vous utilisez. Imaginons le code suivant:

```
String _luke = new String("Luke");  
String dialogue = "Bonjour " + _luke;
```

La dernière ligne est traduite automatiquement par le compilateur de Sun (javac) ou d'IBM (jikes) en :

```
dialogue = new StringBuffer("Bonjour  
").append(_luke).toString();
```

Le compilateur crée donc une instance de la classe StringBuffer avec le premier membre, puis appelle la méthode append() pour y ajouter le second membre. Enfin, la méthode toString() convertit l'objet en instance de String. Ainsi, additionner plusieurs chaînes donnera l'équivalent bytecode de :

```
new StringBuffer("Bonjour  
").append(_luke).append(_chewie).append(_solo);
```

Il faut ici noter que l'addition de constantes sera optimisée par le compilateur pour ne créer qu'une instance de String.

```
String _str = "Login: " + "numéro " + 67;
```

sera interprété de cette manière :

```
String _str = "Login numéro 67";
```

Si vous additionnez un objet différent de String, le compilateur appellera toujours la méthode append() de StringBuffer qui appellera elle même la méthode toString() de l'objet. L'addition d'un type primitif provoquera également l'appel de la méthode append() spécifique. Imaginons maintenant le cas où le premier membre n'est pas une constante String. Le compilateur de Sun traduira ce code :

```
String _str;  
String _luke = "Luke";
```

```
_str = "a" + _luke;  
_str = 'a' + _luke;
```

de la manière suivante :

```
new StringBuffer("a").append(_luke);  
new StringBuffer().append('a').append(_luke);
```

Par contre, jikes créera les instructions :

```
new StringBuffer("a").append(_luke);  
new StringBuffer("a").append(_luke);
```

Jikes possède ici un petit désavantage dans la mesure où il contraint la JVM à créer une nouvelle instance de String même dans l'hypothèse où nous n'ajoutons qu'un seul et unique caractère. Vous pouvez vous même vérifier comment votre compilateur traduit votre code (ces transpositions de String à StringBuffer ayant la fâcheuse tendance à changer au fil des versions des compilateurs) grâce à l'outil javap. Javap est livré avec le JDK et son utilisation est des plus simples :

```
javap -c ClasseAAnalyser
```

Javap accepte également l'option classpath qui s'utilise de la même manière que pour javac ou java. Faites attention: par défaut, javap ne décompile pas les méthodes private. Une analyse minutieuse du listing produit par javap vous servira ainsi à comprendre comment optimiser votre code source.

3. +=, danger !

Comme vous avez pu le constater, les compilateurs font un usage immodéré de la classe StringBuffer. Dans un cadre d'utilisation simple, par exemple lorsqu'il s'agit d'afficher un unique message, l'utilisation de la classe String se suffit à elle même. Mais il est parfois préférable de manipuler directement une instance de StringBuffer. Ceci est particulièrement vrai lorsque la chaîne doit être construite dans une boucle:

```
public String createStringFromArray(Object[] array)  
{  
    String _ret = new String();  
    for (int i = 0; i < array.length; i++)  
        _ret += array[i];  
  
    return _ret;  
}
```

Cette méthode est un très mauvais exemple de l'utilisation de l'opérateur += sur les String. Sachant que l'expression a += b est automatiquement traduite par le compilateur en a = a + b, essayez

de trouver qui se passera si nous additionnons une String et un caractère avec +=. Vous l'avez deviné :

```
_ret = new  
StringBuffer(_ret).append(array[i]).toString();
```

Vous constaterez qu'à chaque passage de la boucle une nouvelle instance de StringBuffer est créée. Or la création d'instance est l'une des opérations les plus coûteuses en termes de performances. La solution serait de pouvoir récupérer l'instance temporaire de StringBuffer créée par le compilateur. Hélas cela n'est pas possible. La solution consiste donc à utiliser uniquement StringBuffer.

```
public void createStringFromArray(Object[] array)  
{  
    StringBuffer _buf = new StringBuffer();  
    for (int i = 0; i < array.length; i++)  
        _buf.append(array[i]);  
  
    return _buf.toString();  
}
```

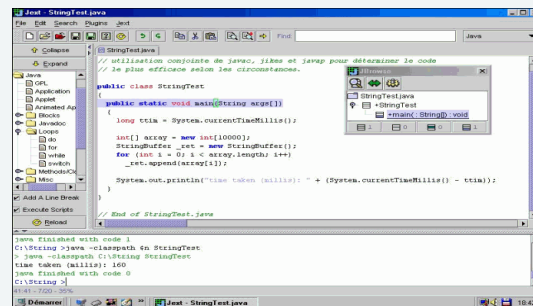
Cette fois-ci, seule une instance de StringBuffer se voit créée. Le gain offert lors de l'exécution est non négligeable. Pour vous donner une idée, avec un tableau de 10 000 objets, la première version de la méthode s'exécute en 14 secondes tandis que la seconde version ne requiert que 170 millisecondes !! Bien qu'impressionnant, ce résultat peut encore être optimisé. En effet, la classe StringBuffer possède un constructeur permettant de définir la taille du tampon contenant la chaîne. La valeur par défaut est de 16. Lors d'un appel à append(), il se peut que le tampon soit plein. En ce cas, un nouveau tampon double est créé et l'ancien est copié dans le nouveau. Lors de la copie, l'instance de StringBuffer contient donc deux tampons: l'ancien et le nouveau de taille double. Avec un ancien tampon possédant déjà une taille conséquente, les exigences en termes de mémoire seront très importantes ! Spécifier la taille du tampon permet d'éviter la copie et donc de perdre du temps et de la mémoire. Dans notre exemple, remplacer la ligne créant le StringBuffer par :

```
StringBuffer _buf = new StringBuffer(array.length);
```

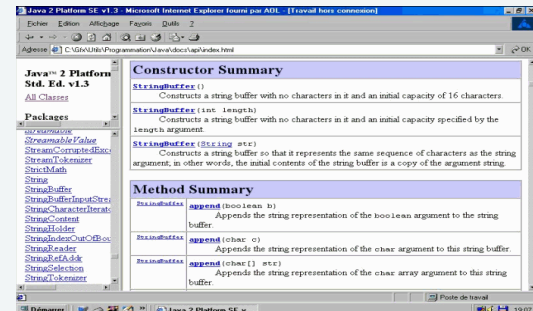
servira à définir une taille de tampon qui ne sera jamais dépassée. Le tampon ne sera jamais copié et l'exécution de la méthode se fera alors en 100 milli-econdes seulement.

Et même si vous pensez rien ne vaut que la classe String pour manipuler du texte, ne vous insurgez pas contre le choix du StringBuffer. En prenant soin de lire la documentation des classes standards du JDK, vous constaterez que StringBuffer possède quelques unes des méthodes les plus utilisées avec les String. En vrac: charAt(), substring(), replace(), length()... Et si par malheur vous devez faire appel à une méthode spécifique à String, la méthode toString() et le constructeur StringBuffer(String str) vous rendront de fiers services. Un autre intérêt du StringBuffer et sa capacité à se voir modifier beaucoup plus aisément que son homologue String. Ainsi, changer un caractère au milieu d'une chaîne requiert de savantes manipulations à l'aide de substring() dans le cas d'une String. A l'inverse, apporter la même modification à la chaîne par le biais d'un StringBuffer ne demande qu'un appel à la méthode setCharAt(). Je ne puis que vous conseiller de lire avec attention la documentation de Sun pour découvrir la puissance de StringBuffer.

4. Dilemme



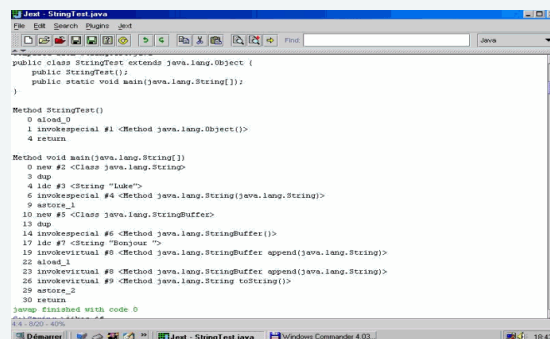
Combien de Strings utilisent ce logiciel ?



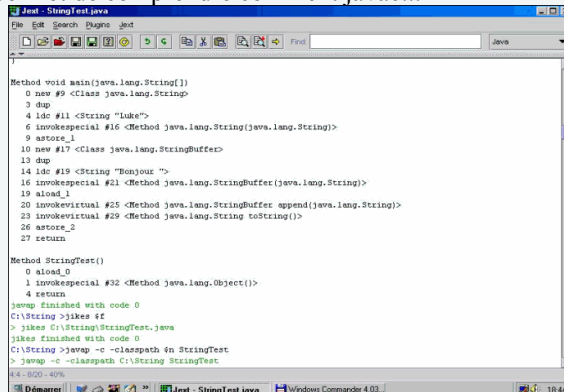
Plongez vous dans la documentation



L'addition de Strings selon les compilateurs



javap permet de comprendre comment javac...



..et jikes traduit votre code

Lire l'article de Romain Guy en ligne : [Lien6](#)

Présentation du framework Google Web Toolkit

Cet article en plusieurs parties est une présentation du framework "Google Web Toolkit".

1. Présentation générale de la solution GWT (*cet article*)
2. Mon premier projet GWT (*à venir*)
3. Comparaison avec SWING et JSP (*à venir*)

1. Introduction

1.1. Remerciements

Merci à [khayyam90](#) et [Yoguy](#) pour leurs relectures attentives

1.2. AJAX, ça nettoie les navigateurs ?

Non, AJAX n'est pas un nouveau produit qui récurer votre cuisine. En fait AJAX c'est plein de chose.

Pour faire simple, c'est un ensemble de technologies WEB, pas forcément récentes, mais qui, mises ensemble, forment un concept qui fait beaucoup parler de lui. Le mieux pour tout savoir est encore d'aller voir là : [AJAX\(Lien7\)](#).

En définitive, AJAX n'est pas un logiciel, n'est pas un framework, ça ne se télécharge pas, mais alors que faire? Comme d'habitude, on va chercher sur Google.

I-C. La réponse de Google



Google a rendu public depuis quelque temps, un framework de développement Web estampillé AJAX.

Ce framework de développement JAVA est utilisé dans plusieurs applications Google comme [Gmail \(Lien8\)](#), [GoogleEarth \(Lien9\)](#) et, je suppose, [Google Notebook \(Lien10\)](#).

Si l'on reprend la définition du [dictionnaire \(Lien11\)](#), la mise en place de l'AJAX dans une application WEB soulève plusieurs problèmes.

Inconvénients :

- ergonomie : l'utilisation d'Ajax entraîne une impossibilité pour l'utilisateur d'utiliser son bouton "Retour" de façon attendue.
- De la même façon, Ajax pose des problèmes pour la mise en place de signets (bookmarks) sur les pages, ainsi que pour l'indexation du contenu des pages.
- temps de latence : les appels vers le serveur peuvent avoir des temps de latence importants qui sont mal perçus et compris par les utilisateurs.
- utilisation de JavaScript : le moteur Ajax fait fortement appel au JavaScript.
- Il faut prévoir pour les utilisateurs ayant désactivé Javascript ou ne pouvant pas l'utiliser, une solution de repli acceptable.
- complexité des développements :
comme tout composant additionnel Ajax offre des possibilités, mais la mise en place peut se révéler coûteuse au moment du développement.

A travers GWT, Google a essayé d'offrir une réponse à ces problèmes.

Pour les temps de latence, il n'y a pas vraiment d'alternative à part proposer, comme dans toute application, un message de patience

(solution retenue dans Gmail).

Pour le reste, GWT offre une solution plus ou moins élégante.

Pour résoudre le problème d'historique entre les pages, Google a mis au point un composant que l'on intègre à notre application sous forme d'IFrame :

```
<iframe id="__gwt_historyFrame" style="width:0;height:0;border:0"></iframe>
```

Cette IFrame va faire en sorte que l'historique des pages soit respecté et permette l'utilisation du bouton précédent du navigateur.

Néanmoins, le principal avantage de ce framework reste la simplicité de développement.

La complexité est réduite du fait d'un développement en JAVA: un seul langage, une seule technologie, on s'extrait avec bonheur du débogage JavaScript.

De plus, GWT fournit une base de classes JUNIT pour tester son application.

Deux modes d'intégration de GWT sont possibles: on peut réaliser une application entièrement avec la technologie GWT ([exemple ICI \(Lien12\)](#)) ou bien intégrer notre composant GWT dans une page Web Standard.

2. Le principe du framework

Le développement d'applications AJAX avec GWT suit un processus simple :

1. Créer votre projet Eclipse avec les outils fournis (projectCreator ou applicationCreator);
2. Créer votre application en JAVA dans Eclipse;
3. Debugger le code avec Eclipse et l'environnement GWT;
4. Compiler votre application avec l'outil Java2Javascript;
5. Déployer votre application.

2.1. Le framework coté zip

Le framework GWT est fourni sous forme d'archive.

Il est développé en JAVA mais l'outil de prévisualisation étant réalisé avec SWT, deux archives sont donc proposées : une pour Windows et l'autre pour le monde Linux.

Pour information, cette différence de plateforme n'impacte que le poste du développeur.

En production, on utilise bien une seule et même librairie quel que soit le système d'exploitation.

Version 1.0.21 (Beta)

May 25, 2006

Platform	Package	Size	MD5 Sum
Windows XP/2000	gwt-windows-1.0.21.zip	13 MB	56b4001cd41826fb1b90bc86cd4fa277
Linux (GTK+ 2.2.1+)	gwt-linux-1.0.21.tar.gz	22 MB	67d00e0e2853c73f92a584873348c559



L'archive contient :

- La documentation et les exemples disponibles sur le site de GWT;
- Le fichier gwt-dev-windows.jar qui contient le compilateur java2javascript, ainsi que l'outil de prévisualisation et de debug;
- Les scripts utilitaires, et les dll pour SWT;
- Et le fichier gwt-user.jar contenant les classes GWT.

2.2. Le Framework coté face

L'utilisation de GWT implique l'utilisation de **JAVA 1.4 maximum** pour le développement de la partie cliente. Le développement coté serveur pouvant se faire en **JAVA 1.5** sans problème. Un petit détail agaçant : l'absence de Javadoc Standard. Le détail des classes est bien fourni mais le format est peu exploitable. Heureusement, des sources documentées sont fournies et rien n'empêche de régénérer soi-même la Javadoc.



Détail intéressant, dans le cas d'une application ne nécessitant pas de dialogue avec le serveur l'ensemble de l'application générée est au format HTML standard. Un simple serveur WEB est donc nécessaire pour publier votre application, et ce sans besoin d'une JVM quelconque.

3. Les outils

Les différentes commandes apportées par GWT pour démarrer un projet sont fournies sous forme de 3 scripts.

Ces 3 scripts n'étant que des appels à des fonctions Java, rien ne vous empêche de les remplacer par un script ANT ou Maven plus adapté à votre IDE préféré.

Quelques plugins sont actuellement en cours de développement pour intégrer GWT à Eclipse, IntelliJ et, dans une moindre mesure, sur NetBeans.

3.1. projectCreator

Cet outil permet de générer le squelette d'une application GWT standard.

De plus, il peut générer les fichiers nécessaires aux projets Eclipse et le script Ant de compilation du projet.

3.2. applicationCreator

Cet outil permet de générer, en plus du squelette, une application GWT minimaliste.

3.3. junitCreator

Cet outil permet de générer un ensemble de classes et de scripts, permettant le test de votre application en mode local et distant.

4. Les licences

GWT est soumis à plusieurs licences dont  [voici le détail](#)

[complet \(Lien14\)](#).

Pour résumer en quelques lignes, l'ensemble du contenu généré avec le framework reste l'entière propriété de son auteur.

À lui de choisir sa licence.

L'ensemble des composants fournis avec le kit (Tomcat, SWT, Xerces, etc.), sont soumis à leurs licences respectives.

L'ensemble des classes fournies dans le fichier gwt-user.jar, classes nécessaires au déploiement en production, sont soumises à la licence Apache V2.0.

Pour le reste, contenu dans le fichier gwt-dev-***.jar, Google n'autorise pas sa commercialisation, sa modification ou sa diffusion sans son autorisation.

Mais le reste, c'est quoi ? Tout simplement le compilateur Java2JavaScript.

Google se réserve donc l'entière propriété et exploitation de son compilateur Java2JavaScript, le reste de la solution étant Open-Source.

Pour faire une comparaison, disons que le modèle se rapproche de celui de SUN avec la JVM.

Il est compréhensible que Google ne souhaite pas voir son produit vendu par un autre, et surtout les bénéfices de cette vente aller à un autre.

5. Et alors ?

GWT serait-il ce framework magique dont rêvent tous les développeurs?

Non, d'abord parce qu'il ne fait toujours pas le café.

Ensuite parce qu'il y a un zone d'ombre suffisamment importante autour du moteur Java2JavaScript, pour avoir quelques méfiances.

Le fait de remettre une couche d'abstraction entre le code écrit et l'application finale est une arme à double tranchant.

L'utilisation de GWT transforme aussi la vision de l'application : on ne conçoit pas une application GWT comme une application WEB standard, mais bel et bien comme on conçoit une application SWING.

L'utilisation de GWT comme base pour un projet dit "non-personnel" doit donc prendre en compte le fait d'oublier une grande partie du code existant (comme pour tout changement de technologie d'ailleurs).

Cela dit, le framework GWT reste selon moi une alternative très intéressante à un développement SWING pour une utilisation multi utilisateur.

Le succès des applications Google crédibilise ce framework et compense le fait que l'on manque pour l'instant de réels retours dans un environnement de production.

Il ne faut pas oublier que le succès d'une application est avant tout lié à ses fonctionnalités.

La question que l'on doit se poser avant de choisir GWT est : "Avons-nous besoin d'AJAX dans notre application ?".

Sans cela, le seul intérêt de GWT reste l'abstraction du code JavaScript et HTML.

6. Liens

Tutoriel : [Introduction à AJAX \(Lien15\)](#) par [Alex Morel \(Lien16\)](#)

Le site de  [Google Web Toolkit \(Lien17\)](#)

Le forum :  <http://groups.google.com/group/Google-Web-Toolkit>

Le blog :  <http://googlewebtoolkit.blogspot.com/>

Retrouvez l'article de Moritan en ligne : [Lien18](#)

[Eclipse Summit Europe du 11 au 12 octobre à Esslingen \(Allemagne\) \(Lien19\)](#)



Prochainement aura lieu le premier événement majeur européen organisé par la fondation Eclipse (il y a certes eu la première édition de [Eclipse Forum Europe](#) récemment).

L'Allemagne, terre d'accueil de nombreuses conférences, accueille donc cet événement de 2 jours. Plus précisément, cette conférence aura lieu à Esslingen, ville voisine de Stuttgart qui, pour ceux qui auraient du mal avec la géographie, se trouve assez proche de la France et de la Suisse.

Au menu de cette conférence en langue anglaise, vous trouverez par exemple des sessions sur :

- le projet Equinox qui répond aux spécifications OSGi (Open Services Gateway initiative)
- le projet BIRT qui n'est plus à présenter
- la plateforme de test TPTP
- le projet RAP (Rich Ajax Platform)
- le projet Corona (collaboration entre projets dans une approche SOA)
- l'intégration de svn avec Subversive
- le projet DALI qui propose un outillage autour de JPA
- Eclipse RCP

Si vous envisagez de suivre cette conférence, n'hésitez pas à réagir ici en décrivant ce que vous en attendez. Peut-être aurez vous l'occasion d'y croiser un membre de la rédaction ...

Petite précision, le coût des 2 jours s'élève à 270 euros.

Retrouvez le billet de Ricky81 en ligne : [Lien20](#)

Un nouveau livre en préparation pour NetBeans

Les livres traitant de NetBeans se comptent sur les doigts de la main.
Mais cela va changer.

Il y a eu celui d'Oreilly couvrant NetBeans 3.3 et qui est sorti lorsque NetBeans 3.4 était disponible. Ce qui fait que la partie couvrant l'API était déjà quelque peu obsolète.

Plus récemment, il y a eu le NetBeans IDE Field Guide qui couvrait NetBeans 4.0 et ensuite NetBeans 4.1. Mais qui ne couvrait que l'utilisation de l'EDI et non l'utilisation de son API. On peut dire que ce livre n'est pas encore trop dépassé si vous utilisez NetBeans 5.x.

Au fait, je rappelle que la traduction en français des brouillons de la plupart des chapitres de ce livre est encore et toujours disponible sur fr.netbeans.org

<http://fr.netbeans.org/servlets/ProjectDocumentList?folderID=29&expandFolder=29&folderID=29>

Mais maintenant, il y a un projet de livre(s) consacré entièrement au développement de modules pour NetBeans. Ce livre couvrira principalement le développement de modules pour l'EDI NetBeans, mais sera également utile pour ceux qui désirent développer des applications Java basées sur la plateforme NetBeans.

GeertJan vous propose un chapitre pilote de ce livre. Très instructif.

http://blogs.sun.com/geertjan/entry/pilot_chapter_for_upcoming_netbeans

Bonne lecture

Vincent

Retrouvez le billet de vbrabant en ligne : [Lien21](#)

Comment vérifier la validité d'un projet Maven ? (Lien22)

auteur : [Eric Reboisson \(Lien23\)](#)

Dans une console de commandes, accéder au répertoire du projet, et exécuter la commande suivante :

Générer le JAR

```
mvn validate
```

Cette commande vérifie que le projet possède toutes les informations nécessaires à son bon fonctionnement avec Maven.

Comment générer le site d'un projet ? (Lien24)

auteur : [Eric Reboisson \(Lien23\)](#)

Dans une console de commandes, accéder au répertoire du projet, et exécuter la commande suivante :

Création du site d'un projet

```
mvn site
```

Le répertoire `/target/site` situé dans votre projet contient alors le site.

Comment connaître l'activité d'un projet ? (Lien25)

auteur : [Eric Reboisson \(Lien23\)](#)

Ajouter au fichier **pom.xml** les balises suivantes :

Configuration du rapport d'activité

```
<project>
...
  <reporting>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-changelog-plugin</artifactId>
      </plugin>
    </plugins>
  </reporting>
...
  <scm>
    <connection>scm:svn:http://chemin_du_projet</connection>
    <developerConnection>scm:svn:https://chemin_du_projet</developerConnection>
    <url>scm:svn:http://chemin_du_projet</url>
  </scm>
...
</project>
```

Le paramétrage du SCM est nécessaire car les rapports d'activité se basent dessus.

Dans une console de commandes, accéder au répertoire du projet, et exécuter la commande suivante :

Générer les rapports d'activité

```
mvn site
```

Le répertoire `/target/site` situé dans votre projet contient maintenant trois rapports d'activité :

- changelog : rapport indiquant toutes les activités sur le SCM.
- dev-activity : rapport indiquant par développeur le nombre de commits, de fichiers modifiés.
- file-activity : rapport indiquant les fichiers qui ont été révisés.

Les formulaires et PHP5

Comment envoyer des informations à un site Web ? Comment est-il possible de publier une news sur son site, voire de gérer tout son site (backend) ?

Ce tutoriel traite de différentes manières permettant de récupérer des informations depuis un formulaire HTML à l'aide d'un script PHP.

1. Introduction

1.1. Remerciements

Un grand merci à [siddh](#) et [titoumimi](#) pour leurs commentaires, ainsi qu'à [Maxoo](#) pour m'avoir fourni la feuille de style utilisée en fin d'article.

1.2. Problématique

Il nous arrive fréquemment de voir des formulaires sur Internet. Ils prennent l'allure d'un sondage, d'une boîte de connexion ou encore d'une interface permettant de publier un message sur un forum.

Ces formulaires permettent à l'internaute d'envoyer des informations au site Web. Ces informations peuvent être enregistrées (cas d'un forum), traitées pour sortir des statistiques (cas d'un sondage) ou encore comprises comme des ordres (panneau d'administration de site). Il est également possible d'envoyer des fichiers sans passer par le protocole FTP.

Les formulaires sont des éléments de la page HTML affichée par le navigateur. Ils ont généralement une destination : le script qui agira lorsque l'internaute validera le formulaire.

La première partie de ce tutoriel, "vision d'ensemble", servira de référence à la suite.

Le reste du tutoriel montrera différentes utilisations de formulaires en les illustrant par des exemples concrets.

Les formulaires ne sont pas complexes mais il y a beaucoup de choses à dire...

Les objectifs du tutoriel

- Voir tous les contrôles HTML des formulaires
- Voir comment traiter en PHP les informations envoyées par un formulaire
- Commencer à voir comment JavaScript peut être utile
- Commencer à voir comment les feuilles de style peuvent être utiles

2. Vision d'ensemble

Je pense qu'il est nécessaire de commencer par voir tous les aspects d'un formulaire. Nous reviendrons par la suite sur les éléments les uns après les autres. Nous allons commencer par les formulaires eux-mêmes, côté HTML, puis nous verrons rapidement comment recevoir les informations d'un formulaire avec PHP.

2.1. Partie HTML : le formulaire lui-même

2.1.1. Obligatoire : la balise <form>

Un formulaire est une balise HTML contenant des éléments permettant à un utilisateur de saisir des informations.

Formulaire minimum : `form.php`

```
<form method="post" action="form.php">
  <input type="submit" value="Envoyer" />
</form>
```

La balise `<form>` est invisible à l'internaute. Elle sert uniquement à indiquer au navigateur Web qu'il a affaire à un formulaire. Nous pouvons voir ici les deux paramètres les plus courants : "method" et "action".

Propriétés de la balise `<form>`

- **action**
Sa valeur détermine l'adresse du script qui recevra le contenu du formulaire, une fois transmis.
Cet attribut est requis.
- **enctype**
Sa valeur détermine le type de données envoyées par le formulaire : sont-ce seulement des informations textuelles ou bien y a-t-il également des fichiers joints ?
Cet attribut est optionnel et vaut "*application/x-www-form-urlencoded*" par défaut. Si l'on souhaite envoyer des fichiers en plus de texte, il faut donner la valeur "*multipart/form-data*".
- **id**
Sa valeur est utilisée pour les manipulations du DOM (Document Object Model) à l'aide d'un script comme ECMAScript (alias JavaScript ou JScript, dans le cas du Web).
- **method**
Les valeurs peuvent être "*get*" ou bien "*post*". Cet attribut est optionnel et vaut "*get*" s'il est omis.

Quelle que soit la *méthode* de soumission choisie, le formulaire enverra ses informations à une page de destination spécifiée grâce à la propriété *action*.

Les deux méthodes de soumission de formulaires

- **GET** : Les variables sont transmises par l'URL, ce qui les rend visibles et modifiables très simplement par l'internaute.
- **POST** : Les variables sont transmises de manière cachée : c'est généralement la méthode préférée des développeurs.



Au cours de ce tutoriel, j'utiliserai exclusivement la méthode POST car c'est la plus courante.

Un formulaire a généralement pour vocation de permettre à l'internaute de saisir des informations. La plus connue des balises permettant d'y parvenir s'appelle `<input>`.

2.1.2. Fondamental : les contrôles `<input>`, `<textarea>`, `<select>` etc.

Les **contrôles** sont les éléments HTML qui permettent de saisir des informations. Il en existe trois différents : *input*, *textarea* et *select*. La balise *input* a une signification très simple : elle permet de "saisir" des informations.

La forme que prend cette balise est définie par la propriété 'type' :

- **text** : Permet d'obtenir une petite boîte de saisie de texte dans la page HTML
- **submit** : Envoie le formulaire
- **reset** : Rétablit le formulaire à son état d'origine
- **radio** : Permet d'obtenir une liste dont un seul choix est possible ; il faut utiliser plusieurs boutons *radio* du même nom (propriété *name*) pour que cela soit utile
- **password** : Même principe que *text*, prévu pour les mots de passe : le texte n'est pas affiché clairement lors de la saisie
- **image** : Même principe que *submit* mais c'est une image au lieu d'un bouton
- **hidden** : Même principe que *text* mais celui-ci n'est pas affiché du tout dans la page
- **file** : Affiche un bouton permettant de sélectionner un fichier de l'ordinateur ; la plupart des navigateurs l'accompagnent d'une case de texte contenant le chemin d'accès au fichier
- **checkbox** : Permet d'obtenir une case à cocher
- **button** : Simplement un bouton ayant la même allure que *submit* ou *reset*, mais celui-ci n'a pas d'utilité propre

Formulaire mixte : form.php

```
<form method="POST" action="test.php" enctype="multipart/form-data">
  text :
  <input type="text" name="text" /><br />

  password :
  <input type="password" name="password" /><br />

  checkbox :
  <input type="checkbox" name="checkbox_1" checked="checked" />
  <input type="checkbox" name="checkbox_2" />
  <input type="checkbox" name="checkbox_3" /><br />

  radio :
  <input type="radio" name="radio" checked="checked" />
  <input type="radio" name="radio" />
  <input type="radio" name="radio" /><br />

  file :
  <input type="file" name="file" /><br />

  textarea :
  <textarea name="textarea" cols="20" rows="7"></textarea><br />

  select :
  <select name="select">
    <option value="Option 1">Option 1</option>
    <option value="Option 2">Option 2</option>
    <option value="Option 3">Option 3</option>
  </select><br />

  <input type="hidden" name="hidden" /><br />
  <input type="image"
  src="http://www.developpez.net/forums/images/logol6.gif" /><br />
  <input type="button" value="Bouton inutile" /><br /><br />

  <input type="submit" value="Envoyer" />
  <input type="reset" value="Rétablir" />
</form>
```

Nous pouvons noter que les contrôles actifs (les boutons) portent une valeur (le texte à afficher), tandis que les contrôles de saisie (les autres) portent un nom. Nous verrons plus tard qu'ils peuvent également porter une valeur. L'élément *textarea* permet de saisir du texte sur plusieurs lignes, ce qui est impossible avec *input type="text"*. Un *select* est une alternative intéressante à des boutons *radio* s'il y a plus de trois choix possibles ou bien si le nombre de choix est dynamique.



Il peut y avoir un ou plusieurs *submit* par formulaire. Tous auront le même effet : soumettre le formulaire au script défini dans la propriété *form action=""*.



Essayez ce formulaire avec la méthode POST puis la méthode GET pour voir la différence.

2.1.3. Facultatif mais pratique : `<fieldset>`, `<legend>` et `<label>`

L'élément *fieldset* permet de regrouper les contrôles par thème. Utilisé conjointement à un élément *legend*, cela permet d'obtenir un formulaire structuré (à la fois dans le code et visuellement dans le navigateur).

xemple de fieldset+legend : form.php

```
<form method="post" action="form.php">
  <fieldset>
    <legend>Informations personnelles</legend>
    Prénom : <input type="text" name="first_name" /><br />
    Nom : <input type="text" name="last_name" /><br />
  </fieldset>
  <fieldset>
    <legend>Informations virtuelles</legend>
    Pseudonyme : <input type="text" name="nickname" /><br />
    Site Web : <input type="text" name="website" /><br />
    Messagerie instantanée : <input type="text"
    name="instant_messenger" /><br />
  </fieldset><br />

  <input type="submit" value="Envoyer" />
  <input type="reset" value="Rétablir" />
</form>
```

Chaque *fieldset* ne contient qu'un seul *legend* mais peut contenir plusieurs contrôles. Il est possible d'assigner un élément *label* à chaque contrôle de la manière suivante :

Exemple de label: form.php

```
<form method="post" action="form.php">
  <fieldset>
    <legend>Informations personnelles</legend>
    <label>
      Prénom : <input type="text" name="first_name" />
    </label><br />
    <label>
      Nom : <input type="text" name="last_name" />
    </label><br />
  </fieldset>
  <fieldset>
    <legend>Informations virtuelles</legend>
    <label>
      Pseudonyme : <input type="text" name="nickname" />
    </label><br />
    <label>
      Site Web : <input type="text" name="website" />
    </label><br />
    <label>
      Messagerie instantanée : <input type="text"
      name="instant_messenger" />
    </label><br />
  </fieldset>
  <fieldset>
    <legend>Fichiers joints</legend>
    <label>
```

```

    Fichier 1 : <input type="file" name="file_1" />
  </label><br />
  <label>
    Fichier 2 : <input type="file" name="file_2" />
  </label><br />
</fieldset><br />

<input type="submit" value="Envoyer" />
<input type="reset" value="Rétablir" />
</form>

```

Cet exemple est identique au précédent à un détail près : si l'on clique sur un label, le curseur est maintenant déplacé dans le contrôle associé.

Il est possible d'associer les labels et les contrôles d'une autre manière (explicitement avec la propriété "for") mais c'est moins intuitif et cela peut entrer en conflit avec certaines de vos habitudes de programmation, dans la mesure où cela fait intervenir la propriété "id" de votre contrôle. Cela permet de placer les *labels* où on le souhaite dans le formulaire. Bien entendu, rien ne vous empêche de répéter les labels.



Exemple d'association explicite : form.php

```

<form method="post" action="form.php">
  <ul>
    <li>Informations personnelles
      <ul>
        <li><label for="first_name">Prénom</label></li>
        <li><label for="last_name">Nom</label></li>
      </ul>
    </li>
    <li>Informations virtuelles
      <ul>

```

```

        <li><label for="nickname">Pseudonyme</label></li>
        <li><label for="website">Site Web</label></li>
        <li><label for="instant_messenger">Messagerie
          instantanée</label></li>
      </ul>
    </li>
    <li>Fichiers joints
      <ul>
        <li><label for="file_1">Fichier 1</label></li>
        <li><label for="file_2">Fichier 2</label></li>
      </ul>
    </li>
  </ul>

  <fieldset>
    <legend>Informations personnelles</legend>
    <input type="text" name="first_name" id="first_name" /><br />
    <input type="text" name="last_name" id="last_name" /><br />
  </fieldset>
  <fieldset>
    <legend>Informations virtuelles</legend>
    <input type="text" name="nickname" id="nickname" /><br />
    <input type="text" name="website" id="website" /><br />
    <input type="text" name="instant_messenger" id="instant_messenger" /><br />
  </fieldset>
  <fieldset>
    <legend>Fichiers joints</legend>
    <input type="file" name="file_1" id="file_1" /><br />
    <input type="file" name="file_2" id="file_2" />
  </fieldset><br />

  <input type="submit" value="Envoyer" />
  <input type="reset" value="Rétablir" />
</form>

```

Retrouvez la suite de l'article de Guillaume Rossolini en ligne : [Lien26](#)

Vu sur les blogs

[Communiqué de presse : Parasoft Jtest 8.0 rationalise le développement d'entreprise Java \(Lien27\)](#)

Jtest 8.0 élargit les possibilités du test fonctionnel Java EE et SOA avec de nouvelles technologies de pointe

Monrovia, Californie - Parasoft Corporation, un éditeur leader de solutions automatisées pour la prévention des erreurs logicielles annonce la disponibilité d'une nouvelle mouture de Parasoft Jtest, sa solution de test unitaire et d'analyse du code Java. Jtest 8.0 offre de nouvelles technologies de test qui aident les équipes à vérifier la fonctionnalité des systèmes d'entreprises complexes et changeants. Elles permettent de réduire les risques de panne et les failles de sécurité pour une satisfaction client accrue. D'autre part, les équipes ont désormais la possibilité de trouver plus de bogues avec leurs ressources existantes, augmentant ainsi leur productivité tout en respectant le budget imparti.

Parmi les nouveautés, on note Bug Detective. En pistant et simulant les chemins d'exécution automatiquement, Bug Detective révèle les bogues à l'exécution qui seraient difficilement détectables lors de tests ou inspections de code manuels. Les utilisateurs peuvent ainsi trouver, diagnostiquer et corriger les catégories d'erreurs logicielles qui échappent à l'analyse basée sur les standards de codage et au test unitaire.

"Nous avons déployé Jtest dans le cadre d'une initiative d'amélioration de la productivité du développement et de la fiabilité du code. Les développeurs doivent maintenant systématiquement effectuer une analyse statique et des tests unitaires sur le code nouveau ou modifié", déclare Laurent Nicolas, Software Development Manager chez Cisco Systems. "Jtest est utilisé pour implémenter les standards de codage et pour détecter et corriger les erreurs tôt dans le cycle de développement. Nous sommes constamment à la recherche de façons d'améliorer nos capacités de détection de bogues et sommes enthousiastes à l'idée d'explorer les nouvelles fonctionnalités de Jtest 8.0."

D'autres nouveautés sont à découvrir avec Jtest 8.0 dont

- La génération et exécution des cas de tests Cactus pour une optimisation des tests en environnement runtime.
- Le Jtest Tracer (ex Test Case Sniffer) - qui enrichit les capacités du sniffer pour améliorer le contrôle sur le "pistage" de l'application à l'exécution et une granularité des cas de tests plus adaptée.

Retrouvez le billet de Miles : [Lien28](#)



Les derniers tutoriels et articles

Signature d'assembly avec .Net

Un élément de base de la sécurité des applications .Net repose sur la signature d'assemblies et l'utilisation de noms forts. L'environnement de développement .Net de base (le SDK) nous propose des utilitaires permettant d'effectuer ces opérations. Même si la signature d'assembly n'est pas un élément de sécurité suffisant, il constitue néanmoins la pierre angulaire de la sécurité d'exécution de code .Net. Au cours de cet article nous verrons à quoi correspondent tous ces procédés de signatures, mais aussi comment les appliquer, quelles sont leurs limites, et ce que l'on peut en attendre concrètement.

1. Introduction

1.1. Clé publique et clé privée ?

On parle de clé publique et de clé privée pour les algorithmes de chiffrement asymétrique. Ces clés permettent de crypter et de décrypter un message, elles seules permettent cette action. La clé publique ne doit pas être gardée secrète et peut être diffusée sans que la confidentialité du message ne soit affectée, par contre, la clé privée, comme son nom l'indique doit rester privée, sa divulgation rendrait inefficace le cryptage, car elle permet de décrypter le message. Pour résumer la clé publique sert à coder un message, alors que la clé privée sert à le décoder. Mais attention, le terme publique et privée sont subjectif, en effet, chacune des clés peut remplir ce rôle, c'est juste que l'on attribue un rôle à chacune, rôle qui bien évidemment n'est plus interchangeable par la suite. La longueur des clés est exprimée en bits, elle va généralement de 128 à 1024 bits suivant le type de clé et d'algorithme. Seuls les algorithmes asymétriques utilisent un jeu de deux clés, les algorithmes symétriques eux, n'utilisent qu'une seule clé. Dans le contexte de la signature d'une assembly, on la signe au préalable avec la clé privée uniquement. La clé publique quant à elle nous servira à contrôler la validité de cette signature.

1.2. Qu'est-ce qu'une signature ?

Une signature électronique permet principalement de garantir que le contenu d'un document n'a pas été modifié après qu'il ait été signé. Appliqué aux assemblies .Net elle garantit l'intégrité du contenu d'origine. Le procédé de signature d'assembly prend donc tout son sens en termes de sécurité et de garantie d'authenticité d'un programme.

2. Qu'est-ce que la signature d'assembly ?

La signature d'une assembly .Net comporte deux composantes fondamentales : hashage de son contenu et un nom fort. Voyons sans plus tarder à quoi correspondent ces deux concepts.

2.1. Strong Names ou Nom Fort

Un nom fort, ou "Strong Names" en anglais (d'ailleurs au cours de cet article nous utiliserons la version française et anglaise sans distinction) correspond à un ensemble d'éléments qui permettent d'identifier avec certitude une assembly donnée. Un nom fort est donc composé des éléments suivants :

- Une version
- Un nom simple
- Une clé publique
- Une clé privée

- Ainsi qu'une culture et une architecture CPU (x86 / x64) optionnelles

Une version est simplement une séquence de quatre chiffres agencés de la sorte :

Numéro majeur de version - Numéro mineur de version - Numéro de "build" - Numéro de révision, cela peut être par exemple "2.7.3.6".

Un nom simple, il s'agit du nom que le développeur a donné à l'assembly, généralement c'est le nom du fichier sans son extension (.dll, .exe,...), il faut donc bien faire attention à ne pas mélanger cela avec un Namespace ou à un nom de classe.

Une clé publique, qui va permettre à l'environnement .Net de vérifier la signature de l'assembly signée avec une clé privée. En effet, une et une seule clé publique correspond à une clé privée. La clé publique sert donc de référent. La clé publique peut être communiquée sans risque. Cette clé publique est toujours intégrée à une assembly signée, et est assez facile à récupérer au sein d'une assembly pour tout développeur .Net aguerri.

Une clé privée, cette clé est celle qui a servi à signer l'assembly, sa valeur doit rester secrète, en effet elle seule garantit l'authenticité et la non corruption du contenu d'une assembly. Sa gestion est donc sensible, mais nous y reviendrons plus tard.



Une culture, pour simplifier il s'agit de la langue et des réglages régionaux, cela peut être US, FR ou autre...

Une architecture de processeur, information en relation avec la machine de destination prévue pour l'assembly, à ce jour Intel (par exemple x86, x64 ou même Itanium)."

Un nom fort est donc un ensemble de procédés et de valeurs qui garantissent l'authenticité et la non corruption d'une assembly. Un nom fort est donc un composant de la technologie .Net qui peut s'avérer extrêmement intéressant dans la gestion de la sécurité. De plus, la signature d'assembly est indispensable pour la gestion du versionning d'assembly ou encore pour placer une assembly dans le GAC (Global Assembly Cache).

2.2. Hashage de l'assembly

En l'état, ce que nous savons de la signature d'une assembly est le fait que l'on utilise un certains nombres de paramètres pour identifier de manière unique une assembly, notamment par le biais

d'une clé publique, ou plutôt d'un jeton correspondant à cette clé publique qui est intégré directement dans l'assembly. Cependant, il manque un élément important de la signature, en effet jusqu'ici rien ne prouve qu'une assembly signée n'ait pas été modifiée depuis qu'elle a été compilée par le développeur. Pour ce faire, un autre procédé va entrer en jeu, il s'agit du hashage. Voyons cela plus en détails.

Tout d'abord le hashage est un procédé mathématique reposant sur des algorithmes permettant de créer une sorte de condensé unique d'un texte ou du moins d'en ensemble de caractères, condensé à partir duquel, il sera absolument impossible de retrouver le texte original, c'est le propre du hashage d'être unidirectionnel et définitif. Pour résumer, après hachage d'une chaîne de caractères on obtient un condensé unique et définitif appelé hash, permettant d'identifier de manière unique et certaine un texte, qui dans notre cas sera le code IL (Intermediate Language) d'une assembly ainsi que certains paramètres de son manifest.

Ce hash est calculé lors de la compilation de notre code (VB.Net, C# ou autre...) en IL, bien évidemment ce hash n'est calculé que si l'on a indiqué au compilateur que l'on souhaitait signer notre assembly. Après calcul de ce hash, celui-ci est encrypté à l'aide de la clé privée, et inscrit dans notre assembly. Dès lors je pense que vous pouvez d'ores et déjà entrevoir la suite... La clé publique qui a été inscrite dans le manifeste de notre assembly, c'est-à-dire dans les métadonnées, va servir à décoder le hash qui y est stocké, et sera comparé avec le résultat du hachage obtenu par l'environnement d'exécution de .Net lors du chargement de l'assembly. Si le hash décrypté par la clé publique contenu dans l'assembly est semblable au hash calculé par l'environnement d'exécution alors l'assembly peut être chargée, dans le cas contraire une exception est levée, et le chargement annulé.

Ainsi grâce à ce procédé de hashage l'on peut déterminer avec certitude si l'assembly a été modifiée ou non depuis qu'elle a été signée par le développeur. Cette certitude repose bien entendu sur la non divulgation de la clé privée qui a servi au cryptage du hash obtenu au moment de la signature.

Au cours de cette partie nous avons pu voir que la signature d'une assembly repose sur un Strong Name qui est un ensemble d'éléments, ainsi que sur le calcul d'un hash de son contenu. La conjugaison des deux va permettre d'identifier avec certitude et de manière unique une assembly, et de garantir que celle-ci n'a pas été modifiée depuis qu'elle a été signée par le développeur à l'aide d'un couple clé publique/clé privée. Même si ce procédé semble être très utile et performant il possède, comme nous le verrons un peu plus tard, certaines limites et contraintes, il participera sans aucun doute au processus de sécurisation du code d'une application, mais il ne peut pas prétendre à incarner la sécurité à lui tout seul.

3. Signer son assembly avec sn.exe

Le SDK du Framework .Net intègre un petit outil en ligne de commande indispensable, il permet de générer le couple clé publique/clé privée RSA permettant de signer votre assembly. Cet outil est très pratique et vous l'utiliserez fréquemment, pour cela nous l'étudierons en détails plus loin dans cet article, ici je me bornerai à vous donner les commandes nécessaires à notre objectif.

3.1. Application exemple

Nous allons utiliser une application console pour toutes les démonstrations de cet article, donc ce sera une application réduite à sa plus simple expression. Il ne vous reste qu'à ouvrir Notepad ou

vos éditeur de texte préféré, et copier le code ci-dessous et enregistrer le fichier sous StrongNamesApplication.vb.

Application exemple

```
Imports System
Imports System.Reflection

<Assembly: AssemblyKeyFile("c:\maCle.snk")>

Module StrongNamesApplication

    Sub Main()
        Console.WriteLine("Application de test de la
signature d'assembly avec des noms forts.")
    End Sub

End Module
```

Comme vous le voyez ci-dessus, l'application console se contente d'afficher un texte lors de son exécution.

3.2. Signature de l'assembly

Lancer donc l'invite de commande de Visual Studio (ou du SDK du Framework .Net), dès lors depuis cette console vous avez accès en ligne de commande à l'outil sn.exe (sn, pour "Strong Names"). Voici la ligne de commande à saisir pour générer la paire de clés RSA :

```
C:\>sn -k maCle.snk
Microsoft (R) .NET Framework Strong Name Utility Version 2.0.50727.42
Copyright (c) Microsoft Corporation. All rights reserved.
Paire de clés écrite dans maCle.snk
```

Un fichier *.snk est créé, ce fichier contient deux clés : une publique et une privée, vous devez prendre garde à ce que ce fichier ne soit pas divulgué, tout du moins la clé privée qu'il contient. Cette clé privée permet de signer l'assembly.

Une fois cette étape passée, il faut maintenant signer notre assembly, cela se fait par le biais du compilateur de votre langage préféré (vbc.exe pour VB.Net, et csc.exe pour C#). Voyons sans plus tarder comment procéder en ligne de commande (la signature d'assembly depuis Visual Studio 2005 sera plus dans la partie suivante de cet article).

```
C:\>vbc /keyfile:c:\maCle.snk c:\StrongNamesApplication.vb
Compilateur Microsoft (R) Visual Basic .NET version 8.0.50727.42
pour Microsoft (R) .NET Framework version 2.0.50727.42
Copyright (c) Microsoft Corporation. Tous droits réservés.
```

Une fois cette commande terminée, une exécutable (StrongNamesApplication.exe) est créée, il s'agit de notre assembly, de notre application. Vous pouvez la lancer en tapant dans l'invite de commande StrongNameApplication.exe, vous voyez alors s'afficher le texte qui doit être affiché par notre application console exemple.

Tout ceci est bien sympathique, mais un peu fastidieux, surtout si vous devez jongler avec plusieurs assembly que vous les recompiliez fréquemment. De plus pourquoi se compliquer la vie si vous possédez un logiciel tel Visual Studio 2005 qui automatise ces tâches. Voici comment procéder pour signer vos assembly depuis Visual Studio 2005.

3.3. La signature plus en détails

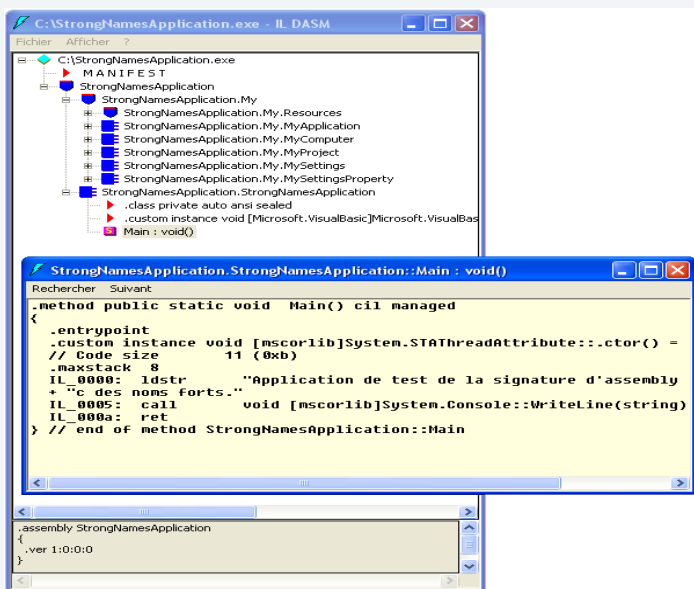
Nous allons utiliser l'outil ILDASM fourni avec le SDK du Framework .Net, il permet de parcourir les assembly pour afficher le code qu'elles contiennent ainsi que les métadonnées, il permet également de désassembler des assembly .Net, et d'afficher le code IL de cette dernière. Ensuite nous utiliserons l'outil ILASM qui lui permet d'assembler du code IL pour générer un *.exe ou une *.dll.

Vous allez me dire quel intérêt pour notre article d'aller voir au coeur de l'assembly ? Tout simplement voir l'incidence directe de la signature, et aussi pour mettre à l'épreuve le processus de vérification de signature des assembly lors de leur chargement par l'environnement d'exécution.

Première étape, le désassemblage de notre dll, nous allons produire un fichier texte contenant le code MSIL.

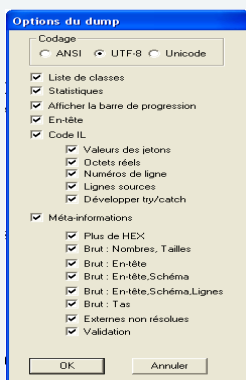
Dans l'invite de commande de Visual Studio 2005, tapez ildasm puis [entrée]. La fenêtre du programme s'ouvre, cliquez alors sur le menu "Fichier", puis "Ouvrir", sélectionnez alors l'assembly.

L'assembly est maintenant chargée dans ildasm, l'on peut parcourir le manifest et ses métadonnées, mais aussi tout le code MSIL.



Désassembleur du SDK : IL DASM

Maintenant, cliquez sur le menu "Fichier", puis "Dump", cochez toutes les options disponibles, laissez utf-8 comme format, et cliquez sur "Ok". Enregistrez le fichier, il aura une extension *.il, pour "Intermediate language". Un fichier du même nom sera aussi créée, mais il aura lui l'extension *.res, c'est un fichier de ressources qui contient des informations relatives à l'assembly que vous avez "dumpé".



Options du dump réalisé avec IL DASM

Dans Visual Studio, ouvrez le fichier avec l'extension *.il qui a été créé par ildasm, il contient les métadonnées et tout le code MSIL de votre assembly.

Tout d'abord si vous avez correctement signé votre assembly et sélectionné toutes les options nécessaires lors du dump, vers la ligne 905/ 910 vous devez trouver la clé publique à qui a été copiée :

Clé publique

```

.publickey = (00 24 00 00 04 80 00 00 94 00 00 00 06 02 00 00 // $.
00 24 00 00 52 53 41 31 00 04 00 00 01 00 01 00 // $.RSA1
0B 59 D3 4B 42 CB F4 69 3C AB D0 60 AE 6D C7 0D // .Y.KB.i<...m.
1B 62 4E 78 A6 30 21 E4 E3 32 6B 3F 25 52 62 E8 // .bNx.0!..2k?%Rb.
66 D0 F2 E0 19 34 91 9E 47 4B 71 35 A5 2A 24 44 // f...4..GKq5.*$D
06 56 B2 FD C5 D9 1E 8B EF 12 0D 00 7D 5C 01 E0 // .V.....}...
58 44 A2 92 D5 5F BE 6E 99 80 D0 22 12 5B AD 30 // XD...n...".{.0
CE 1F 78 61 A8 8D 8B 3E EE DC A0 A6 1F 26 80 0D // ..xa...6.....&
F0 63 9C B1 71 54 99 42 C0 6B 3C 86 47 B7 BA 3F // .c..qT.B.k<.G..?
7D 08 5F E5 A5 EE A6 09 CD 63 AA 17 AA 27 04 BD ) // )._.....c...!..
    
```

(Bien évidemment comme vous avez une clé différente, ce ne sera pas les mêmes valeurs que celles-ci-dessus).

Cette clé, comme je vous l'ai indiqué plus haut, va servir à vérifier la signature de l'assembly et à la valider lors de son chargement par l'environnement d'exécution .Net. En l'état, notre assembly signée doit passer l'étape de la validation avec succès, en effet elle n'a pas été altérée. Cependant pour bien voir que la signature d'assembly n'est pas un gadget, nous allons altérer notre assembly, puis la réassembler, et enfin tenter d'exécuter notre application console pour voir ce qu'il se passe.

Altération de notre assembly :

Sur la ligne après la clé publique, vous devez avoir la ligne suivante

Version de l'assembly

```
.ver 1:0:0:0
```

Remplacez 1:0:0:0 par les chiffres que vous voulez (par exemple 2:1:3:7), ils correspondent au numéro de version de l'assembly. Une fois cette modification effectuée, enregistrez le fichier, et fermez-le.

Dans l'invite de commande de Visual Studio 2005 (ou du SDK), tapez ce qui suit :

Assembler le code MSIL

```
ilasm /EXE c:\StrongNamesApplication.il
```

L'assemblage de notre fichier contenant le code MSIL commence, et doit se terminer par :

Fin de l'assemblage

```
Operation completed successfully
```

Un fichier StrongNamesApplication.exe est alors créé, tentez de le lancer depuis la console. Une exception doit alors être levée, car nous avons altéré l'assembly, et le résultat du hash ne correspond plus à celui effectué lors de la signature de l'assembly par le développeur.

Voici ce que vous devez voir apparaître :



Puis dans la console, l'exception suivante est levée :

La validation de l'assembly à échouée

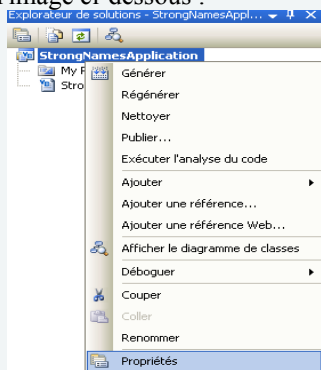
```
Exception non gérée : System.IO.FileLoadException:
Impossible de charger le fichier ou l'assembly
'StrongNamesApplication, Version=1.0.0.0,
Culture=neutral, PublicKeyToken=5ce9c151c1a0a833'
ou une de ses dépendances. Échec de la validation de nom
fort. (Exception de HRESULT : 0x8013141A)
Nom du fichier : 'StrongNamesApplication,
Version=1.0.0.0, Culture=neutral,
PublicKeyToken=5ce9c151c1a0a833' ---> System.Security.
SecurityException: Échec de la validation de nom fort.
(Exception de HRESULT : 0x8013141A)
La zone de l'assembly qui a échoué était :
MyComputer
```

Nous voyons donc que lors de la vérification de la signature l'environnement .Net à détecté grâce au hashage que le contenu de l'assembly à été modifié, il à donc refusé de lancer notre application et a levé une exception.

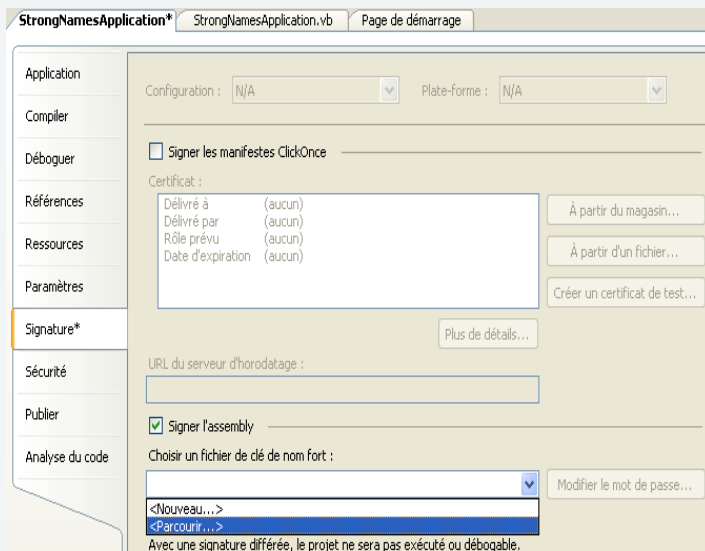
4. Intégration dans Visual Studio 2005

Signer une assembly avec Visual Studio est extrêmement simple, voici la démarche à suivre :

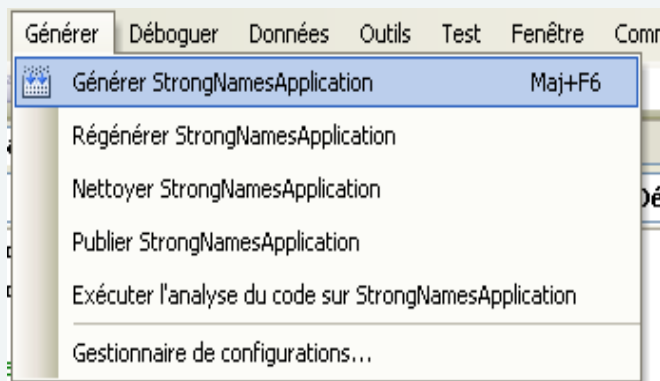
En premier il faut générer une paire de clés avec l'utilitaire sn.exe (voir la commande un peu plus haut dans cet article), puis ensuite, il faut faire un clic droit -> propriété dans l'explorateur de solution comme le montre l'image ci-dessous :



Ensuite aller dans l'onglet signature, cocher la case "Signer l'assembly", puis sélectionner le menu "Parcourir" et pointer vers le fichier *.snk que vous avez généré précédemment.



Votre assembly est dès lors prête pour la signature, cela va être effectué par le compilateur. Pour cela cliquez dans le menu "Générer en haut dans Visual Studio 2005, comme ci-dessous :



Votre assembly est désormais signée, la compilation a réussie, la fenêtre "Sortie" le confirme :

```
***** Génération : 1 a réussi ou est à jour, 0 a échoué, 0 a été ignoré *****
```

Voilà, pas la peine d'en rajouter, c'est aussi simple que cela.

Retrouvez la suite de l'article de Ronald Vasseur : [Lien29](#)

Vu sur les Blogs

[Event] Mercredi du Développement: "C# 2.0, un an après"

Amis développeurs .NET, ce message s'adresse à vous 😊

Vous développez avec C# 2 depuis sa sortie (et même avant) ? Vous pensez maîtrisez ce langage de développement ? Alors dans ce cas, je vous invite à venir voir ce Mercredi du Développement, animé par **Mitsu Furuta**: "C# 2.0, un an après".

Au programme: trucs, astuces, conseils et approfondissement vous seront livrés par cet évangéliste renommé de chez Microsoft France 😊

Pour vous inscrire, c'est par ici:

Rencontre du mercredi – C# 2.0, un an après (niveau 300)

Mercredi 27 Septembre, de 13h30 à 17h00

<http://www.microsoft.com/france/events/event.aspx?EventID=1032309328>

Pour plus d'infos, rendez-vous sur le blog de Mitsu:

<http://blogs.microsoft.fr/mitsufu/archive/2006/09/13/46032.aspx>

A+

Retrouvez le billet de Thomas Lebrun en ligne : [Lien30](#)

Les derniers tutoriels et articles

Gestion dynamique de la lumière avec OpenGL - Partie 1 : l'éclairage par vertex

Ce tutoriel a pour but la mise en place d'un système de gestion des lumières dynamiques par vertex lighting avec OpenGL. nous présenterons aussi une vue d'ensemble des méthodes de gestion des lumières dynamiques dans les moteurs 3D, qui seront traitées pour la plupart dans les prochains tutoriaux.

1. But de ce tutoriel.

Cette série de tutoriaux a pour but d'étudier la mise en oeuvre des systèmes d'éclairage dans un moteur 3D.

Il existe de très nombreuses façons de gérer son éclairage dans un moteur 3D. Dans cette série de tutoriaux, nous aborderons les techniques d'éclairages les plus connus. Néanmoins, nous ne traiterons que les techniques permettant de gérer un éclairage dynamique de nos scènes, et nous ne parlerons que des lumières ponctuelles omnidirectionnelles à rayon fini. Nous n'aborderons donc pas les sources de lumières infini, ou les spots de lumière.

Cette série de tutoriaux se décomposera en 6 parties où nous verrons :

1. L'éclairage par vertex.
2. Le light mapping dynamique simple.
3. Le light mapping dynamique avec texture 3D.
4. L'éclairage par pixel simple.
5. L'éclairage par pixel standard.
6. La création d'un système d'éclairage complexe basé sur les shaders.

Pour la mise en oeuvre, nous utiliserons les bibliothèques OpenGL et GLUT pour gérer l'affichage. Néanmoins, l'ensemble des principes étudié ici est facilement portable sous Direct3D.

2. Pourquoi mettre en place son propre système de gestion de la lumière.

Une question souvent posée sur les forums est de savoir comment gérer les lumières dans un moteur 3D. En effets, même si OpenGL propose une gestion des lumières standard en interne, celle ci n'est pas réellement utilisable dans un moteur complexe. Bien qu'accéléérées par le matériel, les lumières OpenGL posent plusieurs problèmes lorsqu'on souhaite les utiliser dans un moteur complexe :

- L'éclairage est calculé par vertex. Cette façon de gérer l'éclairage oblige à avoir un nombre de polygones élevé pour obtenir une qualité d'éclairage intéressante. En effet, l'éclairage étant interpolé entre les sommets d'un polygone, si les polygones sont trop étendus, l'éclairage ne sera pas bon au centre du polygone. La seule solution pour résoudre ce problème est de découper les grand polygones en plusieurs petits polygones (on parle d'augmentation de la tessellation). Cette solution résout relativement bien le problème de l'interpolation de la lumière, mais elle a aussi comme conséquence d'augmenter fortement la géométrie ce qui a un impact important sur les performances, notamment au niveau de :

- l'envoi de la géométrie à la carte 3D
- l'utilisation de certains algorithmes (par exemple : les collisions) qui seront plus coûteux, ou encore pour le calcul des ombres volumétriques qui dépendent fortement de la géométrie.
- L'éclairage étant calculé par la carte graphique, le nombre de lumières utilisables simultanément est limité par le matériel. C'est principalement cette limitation qui impose la mise en place de systèmes de gestion de la lumière dans les moteurs 3D. En effet, on imagine mal un jeu moderne qui ne permet d'avoir que 8 lumières dans un niveau. Il existe bien entendu des méthodes qui permettent d'activer/désactiver des lumières quand elles ne sont plus visibles. Ceci permet de repousser la limite du nombre de lumières, mais cette limite existe toujours. Imaginez un très grand hangar avec seulement 8 lumières là où on en aurait mis plusieurs dizaines dans la réalité, le résultat risque de ne pas être très crédible.

Tous ces problèmes ont contraint les créateurs de moteurs 3D à mettre en place des systèmes de gestion de la lumière plus ou moins complexes.

3. Les différents modèles d'éclairage.

Il existe de nombreuses manières d'appréhender la lumière. La plupart des systèmes de gestion de la lumière sont basé sur les modèles de Phong et de Blinn. Néanmoins, de nombreux moteurs 3D n'implémentent pas ces modèles dans leurs totalités.



Je ne détail pas ici les modèles de Phong et Blinn, mais le net regorge de sites qui les traitent très bien.

Le principe de base d'un système de gestion de la lumière est de pouvoir noircir plus ou moins les éléments d'une scène en fonction de la position et de l'orientation du pixel par rapport à la lumière.

Dans la plupart des modèles d'éclairage utilisé, la lumière se décompose généralement en 4 composantes :

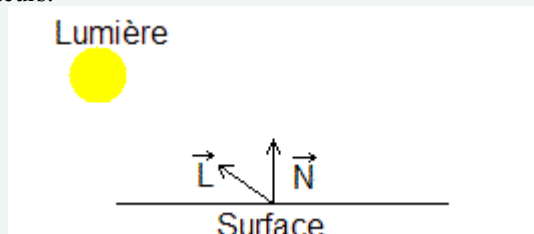
1. La composante ambiante : elle correspond à la luminosité minimum de la scène. Elle permet en fait de simuler les sources de lumières indirectes. En effet, dans la réalité, les objets sont éclairé par 2 types de sources lumineuses : les sources directes et les sources indirectes. Les sources directes sont les générateurs de lumières (ampoule, soleil et autre), et les sources indirectes sont des objets qui ont reçu de la lumière et en restituent une partie. Comme les algorithmes de calcul de restitution de la lumière indirecte ont une complexité temporelle trop

importante pour être utilisé en temps réel, le résultat est approché en utilisant une couleur ambiante commune à toute la scène.

2. **La composante diffuse** : elle correspond à la lumière reçue par l'objet. Elle varie en fonction de la distance et de l'orientation de la surface par rapport à la lumière. Le calcul de la lumière diffuse est généralement

$$\text{diffuse} = N.L * \text{atténuation}$$

Où *diffuse* est la couleur diffuse, *N* est le vecteur normal à la surface, *L* le vecteur allant de la surface à la lumière, *atténuation* est le facteur d'atténuation en fonction de la distance, et le \cdot représente le produit scalaire entre les deux vecteurs.



L'éclairage d'une surface est fonction de la normale à la surface et de la direction vers la lumière



Ce calcul est généralement nommé **NdotL** (dot représentant le produit scalaire) dans la littérature, nous utiliserons donc ce terme au cours des tutoriaux

3. **La composante spéculaire** : elle correspond à la réflexivité de la surface. Par exemple, une surface métallique brillante aura une forte composante spéculaire alors qu'un pneu de voiture n'en aura pas.
4. **La composante émissive** : plus anecdotique, la composante émissive correspond à la lumière émise par la surface. Cette composante est toujours affichée même quand la surface est ombrée.

Généralement, les modèles d'éclairage considèrent que la lumière est additive. Ce qui signifie que si une surface est éclairée par deux lumières, le résultat final sera l'addition des deux éclairages. Ainsi une surface éclairée par une lumière rouge $\{1,0,0\}$ et par une lumière bleu $\{0,0,1\}$ aura la même couleur que si elle avait été éclairé par une lumière violette $\{1,0,1\}$. Ceci est particulièrement important pour déterminer la façon dont seront affichées les lumières par la suite.

Maintenant que nous avons vu cette introduction basique des modèles d'éclairages, nous allons voir des systèmes classiques pour les mettre en oeuvre.

Les types de systèmes de gestion de l'éclairage dynamiques les plus connus sont :

3.1. L'éclairage par vertex

C'est, historiquement, le premier modèle d'éclairage en temps réel utilisé. En effet, l'éclairage en temps réel étant coûteux, lorsque les ordinateurs n'avaient pas de carte accélératrice, il fallait utiliser un système très rapide, même si il présente les désavantages présentés précédemment.

C'est ce modèle dont nous allons étudier la mise en oeuvre dans ce premier tutoriel.

3.2. Le light mapping

Comme son nom l'indique (si vous n'êtes pas anglophobe), le light mapping consiste à stocker l'éclairage dans une texture. Cet algorithme se décompose en 2 étapes :

1. Le rendu de la scène en utilisant la texture d'éclairage. Les parties dans l'ombre sont affichées en noir alors que le reste est affiché dans la couleur de l'éclairage local.
2. Le rendu de la scène avec la texture normale, en multipliant la couleur précédente. Ainsi, si la couleur précédente était du noir, le résultat final sera du noir, donc une zone ombrée. Si la couleur précédente était du blanc, la couleur final obtenu est celle de la texture, donc une zone éclairée.

Cette technique a comme gros avantage de permettre de pré calculer la texture d'éclairage pour les lumières statiques, et, étant basé sur l'utilisation de textures, d'utiliser principalement les ressources de la carte graphique.

Le light mapping est à la base un système de gestion des lumières statiques, mais il est facile de lui ajouter une gestion de l'éclairage dynamique.

Cette méthode sera vue dans les tutoriaux 2 et 3.

3.3. L'éclairage par pixel

Basé sur des extensions comme le DOT3 bump mapping ou les shaders, l'éclairage par pixel est possible de nos jours car les cartes graphiques ayant évolué, le développeur peut maintenant intervenir sur la façon dont sont calculés les pixels avant qu'ils ne soient ajoutés à la scène. Cette méthode de gestion de l'éclairage permet un très bon rendu graphique, tout en étant très flexible. Le choix peut donc être fait entre la qualité graphique ou la performance, en implémentant seulement une partie d'un modèle d'éclairage par exemple.

Cette méthode permettant de nombreuses implémentations, nous l'étudierons durant 3 tutoriaux.

3.4. Autres modèles

Il existe d'autres méthodes (qui sont généralement calculées par pixel) comme les spherical harmonics par exemple, mais ces techniques étant plus anecdotiques dans l'industrie 3D, nous ne les étudierons pas ici.

4. Un premier système de gestion de la lumière : l'éclairage par vertex.

Nous allons maintenant voir comment mettre en oeuvre un système d'éclairage par vertex, mais avant de voir sa mise en oeuvre précise, je vais essayer d'en expliquer rapidement le principe.

4.1. Principe de l'éclairage par vertex

L'éclairage par vertex consiste simplement à effectuer l'ensemble des calculs d'éclairage au niveau des vertex. Ce système est très simple et peut se résumer à l'algorithme en pseudo code suivant :

```
initialiser la luminosité de tout les vertex a la  
lumière ambiante
```

```

pour chaque lumiere
pour chaque face
pour chaque vertex de la face
vertex.couleur = lumiere.couleur * (NdotL) * attenuation
fin pour
fin pour
fin pour
rendre la scene

```

L'algorithme peut être simplifié en ne prenant pas en compte les vertex des faces, mais directement la géométrie, mais cela a pour conséquence de ne pas pouvoir avoir de vertex à normales multiples.

Un vertex à normales multiples est un point de l'espace qui est partagé entre plusieurs faces ayant des orientations très différentes. Par exemple un cube a 8 vertex, mais ses faces ont des orientations suffisamment différentes pour qu'on ne souhaite pas lisser les normales de ces vertex (on parle de normal smoothing), le vertex aura alors plusieurs normales (dans le cas du cube, une par face qui utilise le vertex). Généralement la gestion des normales lissées est laissée à la discrétion de l'artiste qui crée la scène via un outil de modélisation. Ici, nous utiliserons donc les normales contenues dans le fichier.

Pour commencer l'étude de la mise en oeuvre, nous allons voir la structure de données utilisée par nos calculs, ensuite nous aborderons le problème du calcul de l'atténuation de la luminosité, et pour finir, nous parlerons de l'ombrage.

Nous ne traiterons pas des composantes spéculaire et émissive de la lumière, mais leur implémentation est relativement facile à mettre en place.

4.2. Structure de données.

Ici, nous allons parler des structures de donnée utilisé pour le calcul de la luminosité de la scène. J'utilise dans ce tutoriel des classes basiques pour gérer mes vecteurs, couleurs et coordonnées de textures. Ces classes ne seront pas détaillées ici mais elles sont disponibles avec le code.

Pour le tutoriel nous utilisons une scène stockée dans un fichier .obj. Le chargement de ce fichier n'entre pas en compte dans ce tutoriel, mais il nous permet de récupérer les informations de géométrie sous la forme de 4 tableaux qui contiennent :

1. Les positions des vertex.
2. Les normales.
3. Les coordonnées de textures.
4. Les faces.

Les faces sont des structures qui contiennent les indices des vertex/normales/coordonnées de texture de chacun des points de la face. Ici nous ne traitons que des faces triangulaires.

Nous avons donc besoin d'une structure pour stocker ces faces

La structure gerant une face.

```

struct Face
{
    unsigned int vertexIndex[3];
    unsigned int texCoordIndex[3];
    unsigned int normalIndex[3];
    Color color[3];
};

```

Ici, la face contient aussi une couleur pour chacun de ses vertex, c'est la couleur qui sera utilisée pour déterminer l'éclairage final du

vertex. Cette information est propre à ce tutoriel et disparaîtra dans les tutoriaux suivants.

Nous avons besoin d'une classe qui va gérer notre géométrie. Cette classe doit pouvoir stocker les informations sur les vertex, normales, coordonnées de textures et sur les faces. Elle doit aussi permettre de charger un fichier de géométrie, gérer l'éclairage et le rendu de la scène.

La classe gerant un modèle (une scène)

```

class Model
{
private:
    Vecteur * vertex;
    Vecteur * normals;
    TexCoord * texCoords;
    Face * faces;
    int nbVertex;
    int nbNormals;
    int nbTexCoord;
    int nbFaces;

public:
    Model(void);
    virtual ~Model(void);
    bool load(const std::string & filePath);
    void initLighting(const Color& ambient);
    void addLight(Light& light);
    void render();
};

```

La méthode load() sert à charger la géométrie depuis un fichier, elle n'est pas particulièrement intéressante ici, la seule information qui nous intéresse est que le chargement prend en compte les normales partagées, nous avons donc un modèle dont certains vertex ont plusieurs normales et donc un rendu cohérent des cubes et autres formes avec des angles importants entre les faces.

La méthode initLighting prend en paramètre la couleur ambiante et l'assigne à tout les vertex de la scène. Cette méthode donne donc :

La méthode d'initialisation de la lumière

```

void Model::initLighting(const Color& ambient)
{
    for (int i = 0; i < nbFaces; i++)
    {
        faces[i].color[0] = ambient;
        faces[i].color[1] = ambient;
        faces[i].color[2] = ambient;
    }
}

```

La méthode addLight prend en paramètre une lumière, effectue les calculs d'éclairage et les applique aux vertex. Le code donne donc :

La méthode permettant d'ajouter une lumière au modèle

```

void Model::addLight(Light& light)
{
    // pour chaque face
    for (int i = 0; i < nbFaces; i++)
    {
        // pour chaque vertex de la face
        for (int j = 0; j < 3; j++)
        {
            // on calcul la lumiere
            Color c =
light.computeLighting(vertex[faces[i].vertexIndex[j]], normals[faces[i].normalIndex[j]]);
            // et on l'addition avec la

```

```

lumiere precedente
    faces[i].color[j].r += c.r;
    faces[i].color[j].g += c.g;
    faces[i].color[j].b += c.b;
}
}
}

```

Avec `computeLighting`, la fonction de calcul de la luminosité du vertex. Cette méthode sera vue plus loin.

Ici la luminosité calculée est additionnée à celle calculée précédemment, car, comme nous l'avons vue, la lumière est additive.

Il ne nous reste plus que la méthode permettant le rendu de la scène. Elle est relativement simple, et ici, je n'utilise aucune optimisation à base de tableaux de vertex ou autre, j'effectue un simple rendu en mode immédiat.

Méthode de rendu du modèle

```

void Model::render()
{
    glBegin(GL_TRIANGLES);
    // pour chaque face
    for(int i = 0; i < nbFaces; i++)
    {
        // pour chaque vertex de la face
        for (int j = 0; j < 3; j++)
        {
            // on recupere la couleur, les
            // coordonnees de texture et la position
            Color &c = faces[i].color[j];
            TexCoord &tc =
            texCoords[faces[i].texCoordIndex[j]];
            Vecteur &v =
            vertex[faces[i].vertexIndex[j]];
            // et on les envoient à OpenGL
            glColor3f(c.r,c.g,c.b);
            glTexCoord2f(tc.u,tc.v);
            glVertex3f(v.x,v.y,v.z);
        }
    }
    glEnd();
}

```

Maintenant que nous avons notre classe de gestion de la géométrie, nous avons besoins d'une classe qui gère une lumière.

Pour nous simplifier la gestion des calculs, c'est cette classe qui va calculer l'éclairage des vertex en fonction de leur position et de leur normale.

La classe de gestion des lumières

```

class Light
{
private:
    float radius;
    Color color;
    Vecteur position;
public:
    Light(void);
    virtual ~Light(void);
    void setRadius(float r);
    float getRadius();
    void setColor(float r, float g, float b);
    Color getColor();
    void setPosition(float x, float y, float z);
    Vecteur getPosition();
}

```

```

Color computeLighting(const Vecteur & position,
const Vecteur & normal);
};

```

Notre lumière contient simplement une position, un rayon et une couleur. Mis à part les accesseur et modificateur, cette classe ne contient qu'une méthode intéressante : `computeLighting` qui prend en paramètre la position d'un vertex et sa normal et retourne sa couleur d'éclairage. C'est cette méthode qui permet de calculer l'éclairage du vertex comme nous allons le voir.

4.3. Gestion de l'atténuation.

La première chose à prendre en compte dans notre calcul de la luminosité est l'atténuation de la lumière. En effet plus notre vertex est loin de la lumière, moins il est éclairé, et si il dépasse le rayon d'action de la lumière, il n'est plus du tout éclairé.

Il existe plusieurs équations pour calculer l'atténuation (linéaire, quadratique et autre). Ici, nous utiliserons une équation linéaire. C'est à dire qu'un vertex situé à mi-distance du rayon de la lumière recevra la moitié de sa luminosité.

L'équation donne donc

$$\text{atténuation} = \max(0, 1 - (\text{distance} / \text{rayon}))$$

où *atténuation* est le facteur d'atténuation de la lumière en fonction de la distance, *distance* est la distance du vertex par rapport à la position de la lumière et *rayon* est le rayon maximum de la lumière.

Le code du calcul de l'atténuation donne donc :

Code du calcul de l'atténuation

```

// on calcul le vecteur allant du vertex à la lumière.
Vecteur vertexToLight = this->position - position;
float attenuation = std::max<float>(0.0f, 1.0f -
(vertexToLight.getLength() / radius));

```

Nous calculons le vecteur allant du vertex à la lumière. Ici, le sens du vecteur n'a pas d'importance étant donné que ce qui nous intéresse est sa longueur, mais il est néanmoins important de le calculer dans le bon sens car il est réutilisé par la suite dans la gestion de l'ombrage.

4.4. Gestion de l'ombrage.

Ce qui est appelé ombrage en infographie est à différencier des ombres portées. L'ombre portée correspond à l'ombre générée par un objet sur un autre objet. L'ombrage lui est juste le fait qu'une face est moins éclairé si elle ne fait pas face à la lumière et est complètement ombrée si elle tourne le dos à la lumière.

Nous voulons donc obtenir ce comportement pour nos vertex. Pour cela, nous allons justement utiliser le fameux calcul `NdotL`. Mais d'abord un petit rappel de maths sur le produit scalaire s'impose.

Le produit scalaire entre deux vecteurs normalisé (de longueur 1) a une propriété intéressante :

Si les 2 vecteurs sont identiques, le produit scalaire vaut 1. Si les 2 vecteurs ont la même direction, le produit scalaire est compris dans]0..1]. Si les 2 vecteurs sont perpendiculaires, le produit scalaire vaut 0, si ils sont de sens opposé, le produit scalaire est compris dans [-1..0], et si ils sont complètement opposé, le produit scalaire vaut -1.

Appliqué à notre problème, si le produit scalaire entre la normale et

le vecteur vertexToLight est supérieur à 0, c'est que la face est éclairé, sinon, elle est dans l'ombre. Et, encore plus intéressant, si la normal est exactement identique au vecteur vertexToLight (c'est à dire si le polygone fait exactement face à la lumière), le produit scalaire vaut 1. Nous avons donc, à moindre coût, le calcul exacte de l'ombrage de nos faces avec un simple produit scalaire (et une normalisation de vecteur car il faut que le vecteur vertexToLight soit normalisé pour que ce calcul fonctionne). Nous avons donc retrouvé notre NdotL, mais nous l'appliquons par vertex et non par faces.

Notre fonction de calcul de la luminosité devient donc :

Calcul de l'atténuation en prenant en compte l'ombrage

```
// on calcul le vecteur allant du vertex a la lumiere.
Vecteur vertexToLight = this->position - position;
float attenuation = 1.0f - (vertexToLight.getLength() / radius);
// on normalise le vecteur
vertexToLight.normalise();
// on calcul le produit scalaire NdotL avec N = normal
et L = vertexToLight
attenuation *= vertexToLight*normal;
attenuation = std::max<float>(0.0f,attenuation);
```

Maintenant que nous avons calculé la luminosité de notre vertex, nous devons calculer sa couleur. Ceci se fait simplement en multipliant la couleur de la lumière par la luminosité du vertex. Nous obtenons donc le code final de la méthode computeLighting :

La methode de calcul de la couleur d'un vertex

```
Color Light::computeLighting(const Vecteur & position,
const Vecteur & normal)
{
    // on calcul le vecteur allant du vertex à la
    lumiere.
    Vecteur vertexToLight = this->position -
    position;
    float attenuation = 1.0f -
    (vertexToLight.getLength() / radius);
    // on normalise le vecteur
    vertexToLight.normalise();
    // on calcul le produit scalaire
    attenuation *= vertexToLight*normal;
    attenuation = std::max<float>(0.0f,attenuation);
    Color ret = this->color;
    ret.r *= attenuation;
    ret.g *= attenuation;
    ret.b *= attenuation;
    return ret;
}
```

4.5. Le rendu final

Maintenant que nous avons tout ce qu'il faut pour calculer notre éclairage, il nous faut l'afficher. Ceci se fait très simplement en respectant la séquence suivante :

Vu dans la FAQ C

Quelle est la différence entre sizeof(struct MyStruct) et sizeof(struct MyStruct*) ?

Appliquer l'opérateur sizeof à un pointeur de structure fournit la taille d'un pointeur et non celle de la structure.

```
struct data* MyPtrStruct;
```

Ce qui donne le resultat suivant :

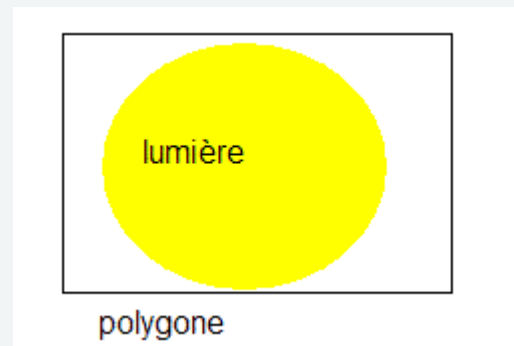


Scène éclairée par 3 lumières : 1 rouge, 1 bleu et 1 blanche

5. Conclusions.

La méthode d'éclairage par vertex est simple à mettre en oeuvre et présente comme avantage important par rapport aux techniques que nous étudierons plus tard de dessiner la scène qu'une seule fois. Nous économisons donc du temps sur l'envoi de la géométrie, mais c'est du coup le processeur qui prend en charge la majorité des calculs.

Un autre défaut vu précédemment est le problème des artefacts d'éclairage sur les polygones trop étendu. Cette artefact se produit principalement au niveau du calcul de l'atténuation. En effet, lorsqu'on place une lumière sur un polygone trop étendu, on peut tomber sur la situation où le rayon de la lumière n'est pas suffisant pour éclairer les vertex du polygone, il n'est donc pas éclairé alors que la lumière le touche.



Cas où l'atténuation de l'éclairage par vertex pose problème. Pour résoudre ces problèmes, nous allons devoir changer de méthode d'éclairage et passer au light mapping que nous verrons dans le prochain tutoriel.

Retrouvez l'article de Michel de Verdelhan en ligne : [Lien32](#) ainsi que la seconde partie de ce tutoriel [Lien32](#)



```
struct data MyStruct

size_t taille1 = sizeof MyPtrStruct; /*
taille d'un pointeur */
size_t taille2 = sizeof MyStruct;
/* taille de la structure */
```

Retrouvez les FAQs de la rubrique : [Lien33](#)

Les derniers tutoriels et articles

Découverte de XGQL

Cet article vise à démontrer la facilité d'utilisation de XGQL pour intégrer des données provenant de sources hétérogènes

1. Qu'est-ce que XGQL ?

XGQL (XML Generator Query Language) est un langage procédural qui permet selon un formalisme XML de générer/transformer des flux provenant de Web Services, de sources de données XML ou de bases de données relationnelles. Il allie ainsi le meilleur des mondes XML et relationnel en utilisant toutes les fonctionnalités de XPath, de XQuery et du SQL.

Pour bien comprendre toutes les possibilités du XGQL, nous allons réaliser une étude de cas assez simple permettant de comparer deux approches :

1. SQL, JAVA et XSLT
2. XGQL

2. Etude de cas

2.1. Contexte

La société *Annu-Artisan* gère un annuaire de sites d'artisans et une liste de leurs produits classés par thèmes.

La société *Annu-Artisan* souhaite proposer un ensemble de sites marchands à thèmes regroupants la production des artisans déjà inscrits dans son annuaire. Comme elle a la capacité de lire et de traiter des flux XML extérieurs, chaque artisan inscrit devra fournir une description de ses produits via un flux XML.

Exemple de flux à fournir

```
<produit>
  <libelle>Marteau</libelle>
  <description_courte>Marteau à enfoncer les clous de girofle</description_courte>
  <description_longue>Un magnifique marteau à enfoncer les clous de girofle : très utile en cuisine</description_longue>
  <images>
    <img>http://photo.ortho.free.fr/images/utills/marteau.jpg</img>
    <img>http://www.stefatelier.com/catalog/images/marteau.jpg</img>
  </images>
  <prix>79</prix>
  <stock>5</stock>
  <delai_livraison>4</delai_livraison>
</produit>
```

Pour simplifier la gestion, les données concernant chaque produit (ID et thème du produit) restent stockées dans l'annuaire de *Annu-Artisan*, dans la table ANNU_PRODUCT ; le champ URL contient l'URL à appeler pour récupérer le flux XML de la description du produit.

IDPRODUITS	IDTHEME	URL
3453	1	http://urlArtisan1/monProduit.xml
892	1	http://urlArtisan2/monProduit.xml
4729	2	http://urlArtisan3/monProduit.xml

Toutes ces données doivent ensuite être normalisées, afin de générer les différentes interfaces des multiples sites.

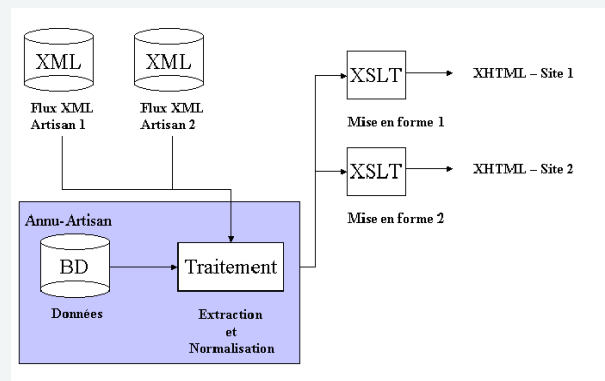


Figure 1 - Présentation du traitement des flux attendus

2.2. Approche SQL/JAVA/XSLT

Un procédé assez simple serait de réaliser quelques servlets agrégatrices des différentes sources de données (base de données et web-service).

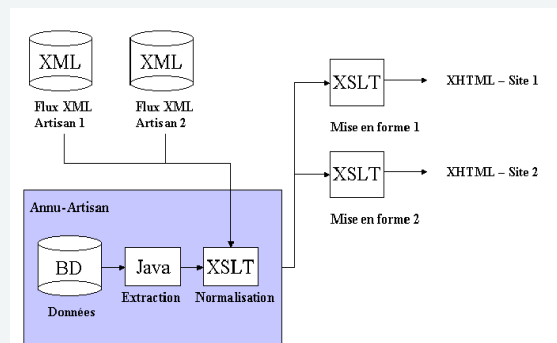


Figure 2 - Traitement des flux avec SQL/Java/XSLT
Le traitement se fait en 4 étapes :

Etape 1 - connexion à la base

Connexion à la base et récupération des informations en fonction d'un thème précis, avec une requête du type : *MAQUERY = "SELECT URL FROM ANNU_PRODUCT WHERE IDTHEME=1"*, le champ URL représentant les différentes adresses des flux XML à récupérer, pour établir une cohérence des données.

```

//Initialisation
ResultSet result = null;
Statement statement = null;
Connection connect = null;
// Connection à la base

try {
    Class.forName(className);
    connect = java.sql.DriverManager.getConnection(url,
name, password);
    connect.setAutoCommit(false);

} catch (Exception e) {
// gestion de l'exception
}

// execution de la requête
try {
    statement =
connect.createStatement(ResultSet.TYPE_SCROLL_INSENSITIV
E,ResultSet.CONCUR_UPDATABLE);
    result = statement.executeQuery(MAQUERY);
} catch (Exception e) {
// gestion de l'exception
}

// fermeture de la connection à la base de donnée
try {
    connect.close();
} catch (Exception e) {
// gestion de l'exception
}

```

Etape 2 - Récupération des informations

Récupération des informations du flux XML en fonction des données intégrées dans la base

```

URL url = new URL(urlFluxXml);
conn = url.openConnection();
conn.connect();

data = new DataInputStream(new BufferedInputStream(
    conn.getInputStream()));

while ((line = data.readLine()) != null) {
    buf.append(line);
}

data.close();

```

Etape 3 - Génération du XML

L'agrégation peut se faire avec un simple objet *StringBuffer*.

```
myXMLResult.append(xmlResult);
```

Etape 4 - Normalisation du XML par un traitement XSLT

Cette feuille XSLT aura pour seul but de supprimer les informations superflues dans le flux XML en fonction des demandes. Par exemple, pour une page de présentation rapide des produits, on voudra uniquement les champs 'libellés' et 'description courte'.

2.3. Réalisation XGQL

En utilisant l'API XGQL, il est possible de générer en quelques lignes de script le XML normalisé souhaité.

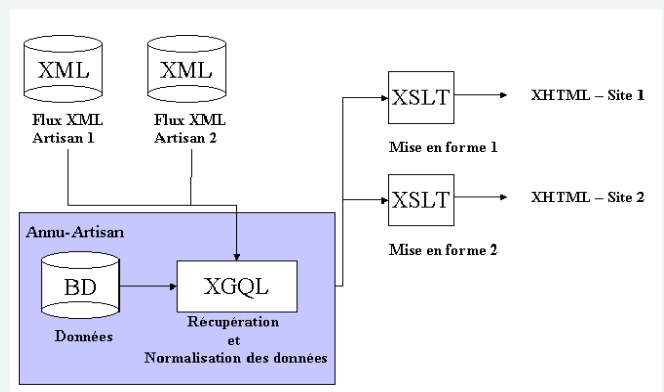


Figure 3 - Traitement des flux avec XGQL

Il est possible de faire le traitement très rapidement de la façon suivante :

```

<?xml version="1.0" encoding="UTF-8"?>

<xgql:root xmlns:xgql="http://www.symeria.com/xgql/">
  <products>

    <!-- definition de la requete -->
    <xgql:var name="select"> SELECT URL FROM ANNU_PRODUCT WHERE
IDTHEME=1</xgql:var>

    <!-- execution de la requete -->
    <xgql:execute name="select">

      <!-- iteration sur les differents résultats -->
      <xgql:row>
        <xgql:var name="productUrl">
          <xgql:column name="url">

via protocole HTTP -->
          <!-- recuperation des differents flux XML,
-->
          <xgql:var name="product">
            <xgql:datasource>${productUrl}</xgql:
datasource>
          </xgql:var>

          <!-- normalisation du flux XML -->
          <produit>
            <xgql:process>document(${product})/pr
oduit/libelle</xgql:process>
            <xgql:process>document(${product})/pr
oduit/description_courte</xgql:process>
          </produit>
        </xgql:row>

      </xgql:execute>

    </products>
  </xgql:root>

```

Ce script génère le résultat suivant :

```

<products>
  <produit>
    <libelle>Marteau</libelle>
    <description_courte>Marteau à enfoncer
les clous de girofle</description_courte>
  </produit>
</products>

```

Sur cet exemple de script XGQL, trois instructions sont particulièrement importantes (et bien pratiques aussi !) :

```

<xgql:execute name="select">
  <xgql:row>...</xgql:row>
</xgql:execute>

```

Exécution d'une requête SQL sur la base, et itération sur les différents résultats de la requête.

```
<xgql:datasource>$productUrl</xgql:datasource>
```

Lecture des données sur un disque local, sur un ftp ou par http (Il s'agit d'une extension du langage présente depuis la version 1.5).

```
<xgql:process>document{$product}/produit/libelle</xgql:process>
```

Interprétation d'une expression Xpath ou XQuery sur une variable XML.

3. Conclusion

Au travers de cette étude de cas, on voit les avantages XGQL :

1. Unification de moyen d'accès, de modification et de transformation pour des flux XML à partir de base de données relationnelles ou données XML
2. Génération de données XML à partir de données relationnelles
3. Syntaxe simple
4. Maintenance facilitée par un code lisible
5. Pas de besoin de recompiler le projet à chaque modification
6. Réduction de la complexité des traitements de transformation (pas de code XSLT nécessaire)

7. Facilement extensible

8. XGQL fonctionne aussi bien sur un Serveur Web qu'en mode API pour des applications ne possédant pas d'interfaces Web

XGQL dispose de plusieurs extensions et est intégré à plusieurs distributions :

- SWAS XGQL : pack regroupant Jetty, HSQLDBD et XGQL correctement configuré et prêt à l'emploi en mode server
- Cocoon XGQLGenerator : Generateur Cocoon pour fichier XGQL
- XGQL:DB2XML : extension de langage transformant une base de données relationnelles en base de données XML

Pour en savoir plus sur XGQL et ses extensions, consultez le site [XGQL \(Lien34\)](#).

4. Remerciements

Je tiens à remercier toute l'équipe responsable de la rubrique XML, et tout particulièrement **GrandFather**, et également **Pierre Martins**, pour son aide.

Retrouvez l'article de Rémi Masson en ligne : [Lien35](#)

Vu dans la FAQ

Comment tester si un élément existe ?

Simplement en mettant le chemin XPath de cet élément dans l'attribut *test* d'un *xsl:if* ou d'un *xsl:when* exemple:

le xml

```
<racine>
<test>existe</test>
</racine>
```

le xslt

```
...
<xsl:if test="/racine/test">
...
</xsl:if>
....
```

Comment effectuer l'équivalent d'une boucle avec compteur en XSLT ?

Il faut écrire une fonction récursive et utiliser les passages de paramètres .

Par exemple, ce code écrit **bonjour!** le nombre de fois spécifié dans le paramètre **fin**

```
<xsl:call-template name="compteur">
  <xsl:with-param name="iteration" select="0"/>
  <xsl:with-param name="fin" select="3"/>
</xsl:call-template>
<xsl:template name="compteur">
  <xsl:param name="iteration"/>
```

```
<xsl:param name="fin"/>
<xsl:if test="$iteration < $fin">
  <xsl:value-of select="'bonjour!'" />
  <xsl:call-template name="compteur">
    <xsl:with-param name="iteration"
select="$iteration + 1"/>
    <xsl:with-param name="fin" select="$fin"/>
  </xsl:call-template>
</xsl:if>
</xsl:template>
```

J'ai en tête de mon document de sortie le prologue XML `<?xml version="1.0" encoding="quelque chose">`, comment l'empêcher ?

Par défaut, si vous ne spécifiez pas d'élément `<xsl:output>` ou que vous ne spécifiez pas ou ne mettez rien dans son attribut **method**, le processeur XSLT va générer du XML en sortie, et ajouter ce prologue. En choisissant judicieusement les valeurs des attributs de `<xsl:output>`, vous avez un ensemble de possibilités.

Pour générer du XML sans prologue (déconseillé, mais parfois nécessaire) :

```
<xsl:output method="xml" omit-xml-declaration="yes" .../>
```

Pour générer du XML sans prologue (déconseillé, mais parfois nécessaire) :

```
<xsl:output method="html" .../>
```

Il y a une autre valeur possible pour **method**, *text*, qui permet d'obtenir une sortie exempte de tout balisage.

Retrouvez la FAQ XML : [Lien36](#)

Les derniers tutoriels et articles

Politique de gestion des droits d'accès à une base de données

Cet article aborde le problème de la gestion des droits d'accès à une base de données par les utilisateurs, les applications et les administrateurs.

1. Introduction à la sécurité

La quasi totalité des applications requièrent une base de données pour le stockage de l'information métier ou comme simple source d'informations.

La sécurité d'une application peut être compromise de multiples manières. Les applications web sont plus particulièrement vulnérables du fait de leur architecture distribuée et leur architecture n-tiers qui multiplient les composants autonomes représentant autant de maillons d'une chaîne de sécurité à rompre.

La compromission d'une application peut engendrer la compromission en cascade des sources de données auxquelles elle a accès. Pour réduire le risque de compromissions en chaîne, il est nécessaire de se doter d'une politique de gestion des privilèges drastique.



Cet article vise un public d'[administrateurs de bases de données \(Lien37\)](#) et de [développeurs \(Lien38\)](#) d'applications accédant à des [bases de données \(Lien39\)](#).

2. Les risques encourus

Les risques propres à une source de données sont les suivants :

- vol de données (perte de confidentialité)
- altération de données (perte d'intégrité)
- destruction de données (remise en cause de la continuité d'activité)
- augmentation du niveau de privilèges d'un utilisateur d'une application (sécurité, espionnage)
- ressources systèmes abusives (déni de service)

Le **vol de données** induit la perte de [confidentialité \(Lien40\)](#) des données stockées. La divulgation de données financières hautement confidentielles peut avoir un impact néfaste sur l'activité d'une entreprise : risque juridique, atteinte à l'image de marque, perte de confiance des partenaires industriels...

L'**altération de données** induit une perte d'[intégrité \(Lien41\)](#), c'est-à-dire que les données ne sont plus dignes de confiance. En fonction de la rapidité de détection et de la qualité des sauvegardes, les conséquences peuvent en être réduites. Mais une application fonctionnant sur des données falsifiées peut voir son comportement fortement influencé : par exemple, un site de commerce électronique pourrait débiter le compte d'un autre client que celui réalisant la commande !

La **destruction de données** remet sérieusement en cause la [continuité de l'activité \(Lien42\)](#) de l'entreprise concernée. Privée de

ses données clients, sans sauvegarde, c'est le dépôt de bilan garanti !

L'**augmentation du niveau de privilèges** d'un utilisateur d'une application est plus insidieux que les risques précédents, car comme pour l'altération de données, il n'est remarqué qu'après un certain laps de temps durant lequel le pirate peut réaliser un grand nombre d'actions malveillantes. Il peut ainsi s'attribuer le droit d'accès à des informations confidentielles, le droit d'accès à des opérations sensibles, voire même prendre le contrôle d'une application.

Selon le SGBD utilisé, des **ressources systèmes** peuvent être attribuées à chaque utilisateur (nombre de requêtes par unité de temps...). Ces ressources peuvent être limitées par l'administrateur système afin d'éviter l'écroulement des capacités de traitement du serveur ([déni de service Lien43](#)) par un utilisateur malveillant. De plus, ceci permet de limiter la portée d'une attaque par altération ou vol de données en limitant le nombre d'opérations réalisables en un temps donné. La conséquence d'un tel risque peut être la paralysie du serveur (perte de [disponibilité Lien44](#)).

On le voit, les risques sont variés et leurs conséquences potentiellement dramatiques. Ainsi, il est nécessaire d'attribuer les droits d'accès avec parcimonie.

3. La politique de gestion des privilèges

3.1. Les types d'utilisateurs

Il faut identifier les utilisateurs ayant besoin d'un accès à la base de données, ils peuvent être de différents types :

- administrateur
- application
- utilisateur

L'**administrateur** est une personne physique ayant tous les droits sur le SGBD, mais pas forcément sur le contenu des bases de données : il peut réaliser des opérations de gestion des droits d'accès et des ressources systèmes mais on pourra choisir d'exclure ou non les droits d'accès en lecture et/ou écriture au contenu des bases de données. Bien que parfaitement logique d'un point de vue métier, pour la protection de données sensibles par exemple, retirer à un administrateur les droits de lecture et d'écriture sur le contenu d'une base de données n'a pas de sens d'un point de vue technique puisqu'il possède les capacités techniques de s'octroyer ses droits là. De plus, les opérations de sauvegarde, de restauration et de maintenance après incident peuvent l'amener à devoir accéder au contenu d'une base de données.

Bref, normalement, c'est l'utilisateur qui a tous les droits sur le

SGBD et les bases de données hébergées. C'est normalement une personne de confiance, compétente et prudente.

Une **application** peut être une application web, un outil de synchronisation entre sources d'informations ou tout programme accédant pour lui-même à la base de données. Ce type d'utilisateur logique n'a rien à voir avec l'utilisateur réel dénotant une personne physique ayant des besoins particuliers. Même si une application est utilisée par des personnes physiques, on pourra choisir de déléguer à l'application la gestion des droits d'accès à l'information en fonction des habilitations qu'elle décide de lui attribuer. Ainsi, une application peut être vue comme un utilisateur de base de données auquel on attribue des droits qu'elle pourra restreindre de façon transparente pour l'utilisateur final de l'application ainsi que pour le SGBD.

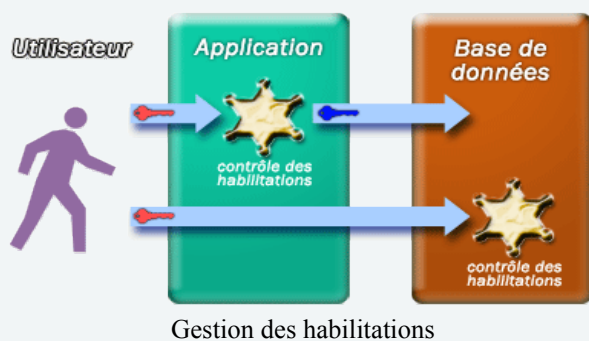
L'**utilisateur** est une personne physique se connectant directement à la base de données [en ligne de commande \(Lien45\)](#) (commande *mysql* sous Linux) ou via une interface graphique (script *phpMyAdmin* sur un Intranet) ou utilisant une application qui va se connecter à la base de données sous l'identité de l'utilisateur (client lourd *MySQL Query Browser*).

3.2. Contrôle des privilèges

La principale question qui se pose lors du développement d'une application, c'est quelle stratégie adopter vis à vis des utilisateurs : contrôle de leurs droits d'accès par l'application ou par le SGBD ?

Par le SGBD. Dans le cas où toute l'information métier repose sur une base de données comportant également toutes les procédures stockées de contrôle de l'intégrité, de la logique métier et des actions utilisateurs, il est logique de déléguer au SGBD le contrôle d'accès et les habilitations. Ceci suppose que l'administrateur de bases de données réalise les opérations d'attribution des privilèges et de synchronisation avec l'annuaire des utilisateurs du système d'information de l'entreprise. L'application ne devient alors qu'une interface graphique ergonomique d'interrogation de la base de données métier.

Par l'application. Dans le cas où l'application gère elle-même le niveau d'accréditation des utilisateurs, elle va se connecter sous sa propre identité logique à la base de données et décider des informations et des opérations que l'utilisateur peut voir, modifier et réaliser. C'est la stratégie employée par les applications dont la logique métier n'est pas intégrée directement dans la base de données et qui gèrent plusieurs sources de données.



3.3. Les privilèges

Il convient pour chaque compte d'accès d'identifier les privilèges minima à accorder ainsi que le niveau de granularité adéquat.



Ici, le terme *utilisateur* désigne une application aussi bien qu'une personne physique.

3.3.1. Classes d'objets et granularité

Les SGBD permettent généralement de spécifier assez finement les privilèges d'un utilisateur en fonction des objets manipuler :

- base de données
- table (relation)
- colonne (attribut)

Ainsi, un utilisateur peut se voir attribuer un privilège pour toute une base de données, ou seulement pour quelques tables, ou encore sur uniquement quelques colonnes de certaines tables.

3.3.2. Classes de privilèges

Les privilèges s'organisent autour de plusieurs classes :

- accès au contenu de l'information
- gestion du schéma de la base de données
- gestion des privilèges utilisateurs
- gestion des paramètres systèmes

Chaque utilisateur devra se voir attribuer des privilèges parmi ces classes.

Le tableau suivant liste les types d'utilisateurs qui devraient être habilités pour chaque classe de privilège :

Classes de privilèges	Types de compte
accès au contenu de l'information	utilisateur, application
gestion du schéma de la base de données	administrateur, application (parfois)
gestion des privilèges utilisateurs	administrateur
gestion des paramètres systèmes	administrateur

3.4. Règles d'attribution des privilèges

Règle fondamentale n°1 : attribution du moindre privilège.

Les utilisateurs ne doivent avoir que le minimum de droits, ceux strictement nécessaires à l'accomplissement de leurs tâches. Les privilèges peuvent évoluer au cours du temps car les besoins et les tâches affectées ne sont pas immuables, mais à un moment donné, seuls les droits indispensables doivent être fournis à un utilisateur.

Règle n°2 : contrôle de la population.

Le personnel d'une entreprise bouge, il y a des départs, des arrivées, des promotions... Les privilèges doivent être synchrones avec la réalité de la population : il faut supprimer les comptes des utilisateurs quittant l'entreprise et de ceux n'étant plus affectés à telle ou telle tâche.

Règle n°3 : supervision de la délégation des tâches d'administration.

Un administrateur peut être amené à déléguer auprès d'une autre personne les tâches d'attribution des privilèges de tout ou partie de la population des utilisateurs. Un contrôle *a posteriori* doit être réalisé afin de vérifier que le résultat de cette délégation est conforme à la politique adoptée.

Règle n°4 : contrôle physique des connexions.

La connexion d'un utilisateur à une base de données peut être réalisée depuis n'importe où dans le monde grâce à Internet. Il est nécessaire de restreindre les connexions à des hôtes spécifiques

connus. Par exemple, le compte d'accès d'une application hébergée sur un serveur devrait voir ses privilèges restreints à l'hôte (ou son domaine) sur lequel elle est hébergée.

Règle n°5 : limitation des ressources utilisées.

Le SGBD offre souvent la possibilité de restreindre les ressources de calcul disponibles pour un utilisateur. Il est recommandé de configurer ces limitations de ressources en fonction de la charge maximale attendue pour un utilisateur. Une personne physique n'a pas besoin de réaliser 100 requêtes à la secondes, mais au contraire, une application gérant elle-même les habilitations peut avoir de gros besoins qui peuvent être cependant limités raisonnablement afin de ne pas compromettre les accès directs en ligne de commande au serveur de base de données...

Règle n°6 : journaliser les comportements suspects.

Certains SGBD permettent de conserver dans des [journaux de log \(Lien46\)](#) les requêtes non conformes aux privilèges accordés à un utilisateur. Il peut être intéressant de les surveiller afin de détecter toute anomalie dénotant des tentatives de piratage.

Règle n°7 : restrictions sur une application en fonction du public.

Une même application web peut avoir plusieurs interfaces différentes selon le contexte d'utilisation : internet / intranet. Par exemple, un magazine de presse en ligne propose une interface de consultation pour tous ainsi qu'un espace d'administration sécurisé. Dans cet exemple, il sera logique d'avoir deux comptes utilisateurs différents à la base de données : le premier en lecture seule sur les tables et colonnes concernant les articles de presse et leurs catégories, le second en écriture. Ainsi, en cas d'attaque du site internet, seules des lectures inoffensives peuvent éventuellement

être réalisées. L'application web choisira le compte d'accès à utiliser selon le contexte d'utilisation.




4. Conclusion

La sécurité des accès à une base de données est une préoccupation de tous les instants. Les privilèges doivent être restreints à l'indispensable et être actualisés régulièrement. Quant aux applications web, vulnérables par essence, elles doivent être développées de manière à réduire le risque d'accès frauduleux aux données.

Une collaboration étroite entre administrateur de base de données et [développeur web \(Lien47\)](#) permet de réduire considérablement les risques liés à la sécurité des bases de données.

5. Voir aussi

Je vous recommande la lecture de ces articles :

-  [SQL : La gestion des privilèges \(Lien48\)](#) de [SQLPro \(Lien49\)](#)
-  [Sécurisation d'une base Oracle \(Lien50\)](#) de [Leo Anderson \(Lien51\)](#)
-  [MySQL 5 : Commande de gestion des comptes utilisateurs \(Lien52\)](#) du *Manuel de Référence de MySQL*

6. Remerciements


Merci à [Xavier Vlieghe \(Lien53\)](#) pour ses remarques constructives.

Retrouvez l'article de Hugo Etiévant en ligne : [Lien51](#)

Vu dans la FAQ Oracle

[Combien coûte la license Oracle ? Est-ce gratuit ?](#)

Tous les produits édités par Oracle sont disponibles en version complète et gratuitement sur [leur site de téléchargement \(Lien55\)](#).

Néanmoins, seul [SQL Developer \(Lien56\)](#) est gratuit quelles que soient les circonstances. En effet, les autres produits ne sont disponibles gratuitement qu'à des fins de **tests du produit** (hors de question de l'utiliser pour tester des développements, seul le prototypage est autorisé) et d'**auto-formation** seulement ! Vous pouvez lire la license sur cette page :  <http://www.oracle.com/technology/software/htdocs/devlic.html> .

Pour tout autre usage professionnel, il faudra vous acquitter d'une license qui dépend de trop nombreux paramètres pour pouvoir être détaillé ici. Je vous invite donc à aller directement sur le site de vente en ligne [Oracle Store \(Lien57\)](#) pour vous faire une idée. Évidemment, il est important de vérifier et/ou discuter des conditions tarifaires avec un commercial Oracle.

A noter également qu'Oracle propose une distribution entièrement gratuite et professionnelle sous l'appellation d'Oracle Database 10g **Express Edition**. Cette version limitée intègre la base de données en version **10g**, le développement d'une application ainsi que du reporting.

[Comment voir les lignes chaînées sur les tables ?](#)

Tout d'abord, il faut analyser la table en question avec la commande suivante :

```
Analyze Table MaTable;
```

```
SELECT table_name,
       owner,
       num_rows,
       chain_cnt,
       (chain_cnt * 100) / num_rows AS ratio,
       pct_used,
       pct_free
FROM   Db_ Tables
WHERE  Owner NOT IN ('SYS','SYSTEM')
       AND table_name = 'MaTable'
       AND Chain_Cnt !=0
ORDER BY 5 desc;
```

Attention, cela ne fonctionne pas avec le package DBMS_STATS.

Retrouvez la FAQ Oracle : [Lien58](#)



Les derniers tutoriels et articles

Firefox et ses meilleures extensions

Sélection et présentation des meilleurs extensions Firefox pour les développeurs.

1. Firefox

Mozilla Firefox est un navigateur tout jeune dont la première version (1.0) fut distribuée le 9 novembre 2004. Ce nouveau navigateur, représenté par un panda rouge, fut (et est encore) développé par une centaine de bénévoles dans un grand projet open-source.

Il a rapidement conquis le public grâce à son moteur d'affichage Gecko, permettant un chargement plus rapidement des pages, mais surtout grâce à une sécurité plus accrue face au géant du marché: Internet Explorer.

Grâce à un développement continu (actuellement en version 1.5.0.6) et à l'ajout d'une infinité de fonctionnalités, Firefox continue de séduire les gens et gagne peu à peu des "parts de marché".

Bien qu'il contienne beaucoup de fonctionnalités que ses concurrents n'ont pas, il possède également la plus large collection d'extensions, lui permettant une infinité de fonctionnalités.

C'est justement dû à ce (trop) grand nombre d'extensions que la rédaction de [Developpez.com](#) a sélectionné, pour vous développeurs, les meilleures extensions.



Si vous désirez tout savoir sur Firefox, je vous invite à lire [sa page Wikipedia \(Lien59\)](#)

Téléchargement: [ici \(Lien60\)](#)

2. Les extensions

2.1. Qu'est-ce qu'une extension?

Même si Firefox possède un grand nombre de fonctionnalités, dont certaines sont enviées par d'autres navigateurs, il n'en reste pas moins que Firefox ne répond pas à toutes les exigences des utilisateurs.

C'est pourquoi il est possible d'installer des extensions. Chaque extension est un module qui s'intègre au navigateur, soit par un

bouton, soit par un panneau latéral, soit de façon invisible, et fournit alors des fonctionnalités supplémentaires.

Ces fonctionnalités peuvent être:

- des options de configuration supplémentaires
- un client ftp ou mail intégré
- des options d'affichage supplémentaires
- des générateurs de mot de passe
- d'autres fonctionnalités moins utiles
- etc...

2.2. Les meilleures extensions

Avant d'aller plus loin et de se faire remonter les bretelles, nous tenons à signaler que les extensions citées ici, ne sont pas les meilleures dans leur qualité de fonctionnalités, de code ou d'intégration, mais simplement les extensions qui ont pour nous, le plus grand intérêt et qui devraient servir à un maximum de personnes.

Qu'importe les commentaires que nous ferons sur les extensions citées, nous tenons tout d'abord à féliciter les auteurs de ces extensions pour le travail fourni gratuitement à la communauté de Firefox.

2.2.1. Adblock

Qu'est ce qui représente le plus gros fléau de l'internet aujourd'hui et qui fait vivre des milliers de personnes? non ce ne sont pas les virus mais les publicités. Sur cette pollution visuelle soi-disant ciblée, a fait la fortune de grosses sociétés (g**gle par exemple) et est dans bien des cas non désirée par la personne naviguant sur un site web.




Le webmaster est en droit d'afficher de la publicité, c'est souvent sa seule source de revenu, et permet au site de rester en ligne. Néanmoins, le lecteur est lui aussi en droit de ne pas lire la publicité et peut même décider de ne pas l'afficher. C'est avec cette mentalité que *the Adblock Crew* a créé l'extension Adblock qui permet sur les sites de son choix, de définir (via un simple clic droit) les publicités (images, GIF animés, flash, etc) que l'on ne souhaite plus voir s'afficher. Ainsi, à chaque fois que vous retournerez sur un site où vous avez déjà bloqué une pub ou une simple bannière que vous estimez trop lourde pour votre bande passante, cette dernière ne se réaffichera pas et vos chargements de pages en seront augmentés.



Mais voilà, ce n'était pas suffisant et au vu de certains bogues de l'extension Adblock et surtout le manque de fonctionnalités, Wladimir Palant décida de reprendre le développement de l'extension pour créer Adblock Plus, la petite sœur d' Adblock proposant parmi tant d'autres, une liste blanche et une meilleure compatibilité avec G.

Qu'est-ce donc que ce G. et à quoi sert-il? Et bien, il met tout simplement votre Adblock à jour avec une liste de pubs prédéfinies. En effet, avec Adblock plus, vous deviez bloquer (une fois) chaque pub que vous rencontriez mais si vous arriviez sur un nouveau site web ou si la publicité changeait, vous deviez réitérer la manipulation. Heureusement, ce n'est plus le cas car des âmes charitables ont décidé de créer "l'extension" de l'extension Adblock:

FilterSet G. for Adblock.

Nous déconseillons personnellement d'utiliser  [Adblock \(Lien61\)](#) et d'utiliser  [Adblock Plus \(Lien62\)](#) à la place. Nous conseillons également l'installation de l'extension  [FilterSet G. for Adblock \(Lien63\)](#)

2.2.2. IETab

Quel est le plus gros avantage de Firefox? Il n'est pas IE. Et son plus gros défaut? Il faut parfois revenir à IE pour avoir un site fonctionnel.

En effet, lorsque votre site ne marche pas correctement sous Firefox, vous devez copier/coller l'url dans une fenêtre d'Internet Explorer. Cette manipulation peut devenir rapidement fatigante. Heureusement pour nous il existe l'extension IEView, qui permet d'afficher dans un menu contextuel (bouton droit), une option pour ouvrir automatiquement l'url dans une page IE.

Cette option a pendant longtemps été la solution la plus rapide et la plus efficace mais a néanmoins ses inconvénients comme le fait d'avoir des fenêtres Firefox et IE ouvertes simultanément et rend plus difficile la possibilité de retrouver rapidement sa page. Une solution a alors vu le jour, grâce à l'extension IETab. Ceux qui l'ont testé ne peuvent plus s'en passer. Cette dernière charge un composant de Windows (utilisé par IE) pour afficher un "onglet IE" directement dans Firefox.

Premier avantage: une seule fenêtre et plusieurs onglets FF ou IE possibles

Deuxième gros avantage de cette extension: son switch. L'installation de cette extension installe deux boutons dans Firefox (affichage configurable) qui permettent de switcher entre le moteur FF et le moteur IE. Ainsi, si vous êtes en train de naviguer sur www.toto.com et que cela ne marche pas comme prévu, il vous suffit de cliquer sur le bouton pour que l'onglet se transforme en onglet IE et charge la page comme si vous étiez sous IE.

Troisième avantage: sa configuration. Il est en effet possible de configurer l'extension pour ouvrir automatiquement certaines URLS. Ainsi, si vous configurez l'url www.toto.com, chaque fois que vous taperez son adresse dans la barre d'adresse, l'onglet se transformera automatiquement onglet IE.

Pour les désavantages: la sécurité. En effet, IETab charge un onglet avec le moteur IE, avec tout ce qui va avec: menu contextuel d'IE, bugs, failles, etc. Ajoutons à cela que les autres extensions de Firefox ne marcheront pas forcément sur cet onglet.

Téléchargement: [ici \(Lien64\)](#)

2.2.3. GetJetableMail

voici une extension peu souvent connue, j'ai nommé GetJetableMail, ce qui, pour les anglophobes, veut dire: obtenir un mail jetable.

Qu'est-ce que signifie un mail jetable? Et bien comme son nom l'indique, il consiste à obtenir un mail qui ne marchera plus au bout d'un temps T défini par l'utilisateur.

Mais pour quel intérêt? le spam pardi!!! Si vous êtes un utilisateur intelligent, vous possédez forcément un compte mail qui sert de mail "poubelle", et avec lequel vous vous inscrivez sur les sites pour lequel un compte "bidon" est demandé, et qui sont susceptibles d'être source de mails de spam.

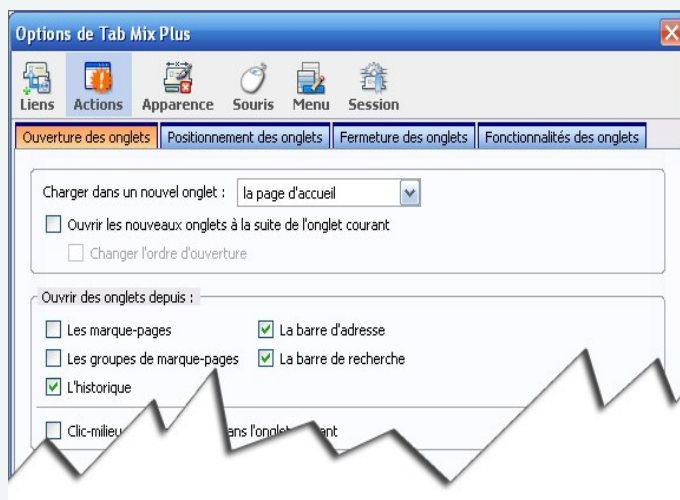
Mais cette solution est-elle optimisée? la réponse est non! Parce que rapidement cette boîte poubelle peut se retrouver avec une 40aine(ou plus) de spam par jour. Il existe alors l'ultime solution avec GetJetableMail.

Fonctionnement: ce dernier crée une adresse bidon qui redirige les mails vers votre boîte mail puis se détruit au bout d'un certain temps (24h par exemple). Ainsi, si vous aviez une adresse toto@dvp.com, vous créez une adresse GetJetableMail (auto générée par le plugin. exemple: elj3on98d3rin9v8@jetable.com) et vous saisissez le mail dans le formulaire du site sur lequel vous désirez vous inscrire. Ainsi les mails seront envoyés à cette adresse qui renverra automatiquement vers toto@dvp.com. 24h plus tard, l'adresse étant détruite, vous ne recevrez plus les mails du site web.

Téléchargement: [ici \(Lien65\)](#)

2.2.4. Tab Mix Plus

Nous ne décrivons pas toutes les options de Tab Mix Plus, mais nous le désignerons comme de très loin, le plus avancé des customizers des paramètres de navigation par onglet. Il vous permettra de paramétrer très finement chaque ouverture, fermeture, et action sur les onglets.-



Téléchargement: [ici \(Lien66\)](#)

2.2.5. ScrapBook

Scrapbook est une extension que vous ne trouverez que sur le site de son(ses) auteur(s):

Ne vous est-il jamais arrivé de cliquer sur l'un de vos anciens favoris et de voir que la page demandée n'existe plus? Ceci est maintenant fini avec Scrapbook. Son fonctionnement, au lieu de simplement garder un lien, il aspire et enregistre la page. Mais ça, nous pouvions déjà le faire, ça remplissait juste les dossiers un peu n'importe comment.

Dorénavant, vos pages seront rangées correctement dans le panneau latéral, selon l'arborescence de votre choix.



Mais ce n'est pas tout, les pages sauvegardées sont éditables via une petite barre d'outils qui vous permet d'éditer ou supprimer certains éléments de la page que vous ne souhaitez pas garder.

Téléchargement: [ici \(Lien67\)](#)

2.2.6. Web Developer

Cette extension est LA trousse à outils du développeur Web. Ses fonctionnalités se comptent par dizaines. Il gère aussi bien tout ce qui est cookie, formulaire, css, JavaScript, validation W3C, test en plusieurs résolutions. Voici un descriptif plus détaillé des menus de cette barre d'outils:

Disable

Permet de désactiver de nombreuses fonctionnalités: le cache, java, JavaScript, les couleurs de la page, etc

Cookies

Fournit différents outils sur les cookies avec entre autres, l'ajout de cookie, la visualisation du cookie du site en cours, la désactivation des cookies.

Css

Fournit différents outils pour travailler sur les CSS: leur affichage, l'édition en temps réel et plein d'autres choses.

Forms

Permet de travailler avec les formulaires des pages web. Afficher les détails de chaque formulaire, voir les champs cachés, remplir les champs, etc

Images

Permet d'afficher et de travailler sur les images de la page. Les cacher ou au contraire afficher leurs informations (taille, chemin de l'image, dimensions, texte alternatif, etc)

Information

Fournit différentes informations sur le document web en cours. Permet également de faire ressortir certains éléments (blocks) de la page pour une meilleure visibilité.

Miscellaneous

Permet principalement d'éditer le code source HTML et de voir les modifications en temps réel

Outline

Permet de surligner les éléments de son choix.

Resize

Permet de redimensionner la fenêtre pour faire des tests de résolution ou encore d'afficher la taille courante de la fenêtre pour des résolutions "exotiques"

Tools

Fournit un grand nombre d'outils en tout genre passant de la validation HTML ou des CSS, jusqu'à l'utilisation d'une console JavaScript pour détecter les erreurs, etc

View Source

Permet d'afficher soit la source entière, soit simplement la source des éléments générés (PHP, ASP, etc) ou encore d'afficher la source dans un programme externe.



Bref, un must-have pour tout développeur web.

Téléchargement: [ici \(Lien68\)](#)

2.2.7. Gtranslate

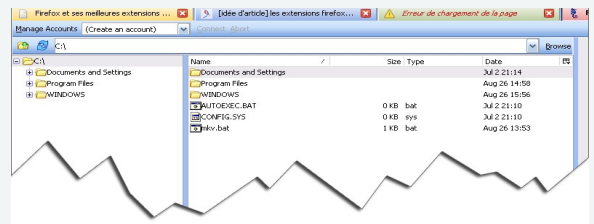
Vous venez de tomber sur un mot qui ne vous dit rien, et vous avez besoin de le traduire. Mais sélectionner le texte, trouver un bon site de traduction, traduire puis revenir à la page originelle peut être fastidieux, surtout si plusieurs mots de la page sont à traduire. C'est fini avec cette extension. Vous présélectionnez la conversion de votre choix et vous n'avez qu'à faire un bouton droit sur votre sélection pour le voir traduire.



Téléchargement: [ici \(Lien70\)](#)

2.2.8. FireFTP

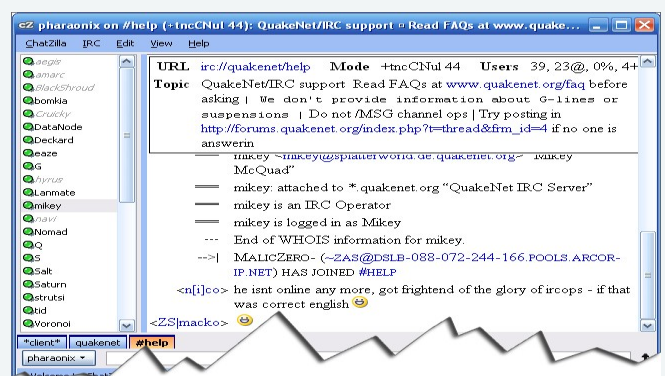
FireFTP est un client ftp minimaliste mais néanmoins complet qui s'ouvre dans un onglet et vous permettra d'accéder à des FTP pour y transférer tout type de fichiers. FireFtp permet de gérer plusieurs comptes mémorisables, de choisir le type de transfert, de passer par un proxy, etc.



Téléchargement: [ici \(Lien71\)](#)

2.2.9. ChatZilla

Il puisse arriver qu'un développeur ait besoin d'aide (ou simplement de détente) et par conséquent, qu'il puisse obtenir des réponses très rapidement. Dans ce cas, le chat IRC est souvent la solution la plus adaptée mais nécessitant un client IRC. ChatZilla est alors le logiciel qu'il vous faut. Client IRC minimal, il vous permet d'utiliser très facilement tout chat. A vous les programmes solutionnés rapidement et si cela ne vous suffit pas, il vous reste le chat de developpez.com: <http://chat.developpez.com>



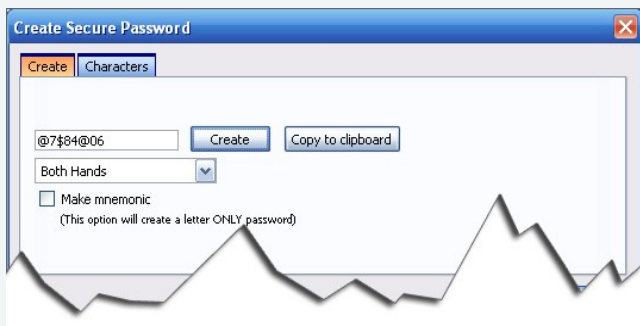
Téléchargement: [ici \(Lien72\)](#)

2.2.10. SecurePassword

Bon nombres d'informaticiens sont incapables d'avoir les notions de base de la sécurité et l'utilisation de mot de passes "sécurisés". On voit même des administrateurs système avec des mots de passe comme "soleil" ou "pamela"...

Heureusement, il y a ceux qui savent qu'il faut créer des mots de passe alphanumériques. Malheureusement pour eux, la génération d'un mot de passe aléatoire par le cerveau humain n'est pas toujours facile.

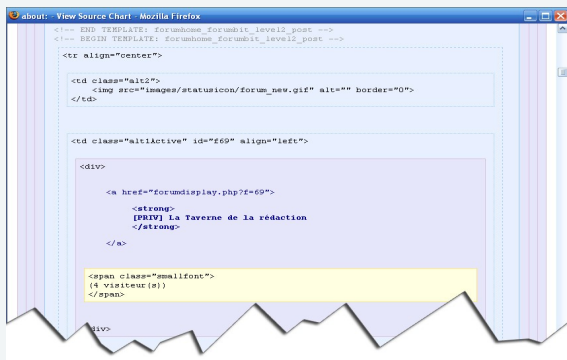
C'est là que cette extension entre en jeu. Entièrement paramétrable, cette extension permet de générer en quelques secondes un mot de passe totalement aléatoire. Fini les mots de passe trop faciles, à vous les mots de passe difficiles à retenir :)



Téléchargement: [ici \(Lien73\)](#)

2.2.11. View Source Chart

Cette petite extension permet d'afficher le code source d'une page de façon aérée et très claire grâce à un embriquement de cadre colorés. Il vous permettra de lire plus rapidement tout type de code source, même le plus sale du monde :)



Téléchargement: [ici \(Lien74\)](#)

2.2.12. FireBug

FireBug est l'extension parfaite pour le développement qui touche au javascript (y compris AJAX donc). Il contient un inspecteur de code qui détecte les erreurs mais surtout le débogueur qui permet

d'avancer pas à pas dans le code javascript afin de mieux comprendre les différentes étapes des fonctions pour y découvrir un éventuel comportement inhabituel

Téléchargement: [ici \(Lien75\)](#)

2.3. Les autres extensions

Bien d'autres extensions sont intéressantes mais il ne nous a pas été possible de les citer toutes. Néanmoins parmi les extensions ayant aussi retenu notre attention, nous pouvons parler de

customizegoogle (Lien76): enlève les pubs de google et plein d'autres choses

autohide (Lien77): cache les barres d'outils qui réapparaissent au passage de la souris

pdf_download (Lien78): permet de télécharger le PDF et de l'afficher dans un programme externe (évite de faire ramer Firefox voire le faire planter)

Menu Editor (Lien79): permet de choisir tous les menus que vous n'utilisez pas et que vous ne voulez pas voir apparaître

External Application buttons (Lien80): Cette extension est la copie d'une fonctionnalité du navigateur Maxthon (et peut-être d'autres). Elle vous permettra de mettre des raccourcis vers des applications externes de votre choix. Ainsi vous pouvez mettre un raccourci vous permettant d'accéder rapidement à votre éditeur de code préféré, puis à votre client FTP (si vous n'utilisez pas FireFTP) et ce, sans sortir de Firefox.

Bookmarks synchronizer (Lien81): permet d'avoir vos favoris en ligne, ce qui permet de les avoir directement dans Firefox mais sur n'importe quelle machine.

Je vous laisse néanmoins découvrir les centaines d'extensions qui existent afin que vous vous fassiez votre propre idée: [Geckozone \(Lien82\)](#)

3. Conclusion

Firefox est un tout jeune navigateur qui en deux ans a su démontrer sa puissance, sa sécurité et ses fonctionnalités de base. Grâce à des milliers de bénévoles et leurs extensions, il est actuellement le navigateur web le plus complet et le plus personnalisable.

4. Remerciements

Remerciements tous particuliers aux auteurs de ces extensions et également à ceux des autres extensions qui, par leur temps libre dépensé, nous facilite notre vie de tous les jours. Remerciements également à tous les membres de la rédaction ayant participé à la sélection des extensions et l'écriture de cet article, à savoir: [netah25](#), [gangssoleil](#), [vow](#), [gnux](#), [cchatelain](#), [2Eurocents](#), [gege2061](#) et surtout [gretch](#).

Je tiens également à remercier [ArHacKnIde](#) pour les corrections apportées à l'article.

Retrouvez l'article de Louis-Guillaume Morand en ligne : [Lien83](#)

Vu dans la FAQ

Quelle est la configuration minimale pour faire fonctionner Windows Vista ?

Actuellement, aucune configuration minimum n'a été annoncée officiellement, néanmoins il est facile de se faire une idée.

- Le processeur doit avoir une fréquence minimale de **800MHz**.
- Contrairement à ses prédécesseurs, les ressources mémoire seront plus exigeantes et **512Mo de ram** seront demandées mais 1Go de ram ne sera pas superflu.
- **15 Go** d'espace disponible sur le disque dur.
- une bonne carte graphique *

- Lecteur CD ou DVD.

- La carte graphique sera le point central de la configuration de Windows Vista où pour celle-ci, il sera conseillé d'avoir un minimum de 128Mo (même si 64Mo seront suffisant). la carte graphique devra être compatible avec les librairies DirectX 9

Retrouvez la FAQ Vista en ligne : [Lien83](#)

Les derniers tutoriels et articles

Utiliser la librairie Windows Image Acquisition en VBA

Ce document présente l'utilisation de la librairie Windows Image Acquisition Automation Library v2.0 ([WIA](#)).

Cette librairie permet de manipuler les images.

Les informations et les exemples proposés sont en partie issus des samples VB6 fournis lors du téléchargement de la librairie (wiaaut.dll), et adaptés pour une utilisation en VBA Excel.

1. Préambule

Tout d'abord, voici un résumé des manipulations possibles:

- * Accéder aux propriétés d'un fichier image.
- * Lire et modifier les fichiers image, ainsi que leurs propriétés.
- * Manipuler les images: Rotation, Retournement, Rognage, Fusion, Compression, Conversion, redimensionnement, Décomposition des frames d'une image (par exemple d'un GIF animé)...
- * Visualiser et gérer les images des webCams, appareils photos numériques et scanners connectées à votre PC.

La Bibliothèque WIA est principalement conçue pour gérer les formats d'image PNG, BMP, JPG, GIF et TIFF.

Les autres formats sont à tester en fonction de votre système d'exploitation.

Téléchargez la librairie si elle n'est pas installée sur votre poste:

 [Le lien sur le site Microsoft \(Lien84\)](#)

Et suivez les conseils d'installation fournis dans le fichier d'aide **ReadMe.txt**



La configuration minimale nécessaire: **Windows XP**

Les exemples suivants ont été Testés avec WinXP et Excel2002 / Excel2000.

2. Manipuler les images

2.1. L'utilisation des filtres

Ce chapitre décrit les filtres disponibles et présente quelques exemples d'utilisation.

2.1.1. Lister tous les filtres disponibles et afficher une description pour chacun d'entre eux

Vb

```
Sub listerInformationsGeneralesFiltres()  
    Dim x As Integer  
    Dim IP As ImageProcess  
  
    'creation du gestionnaire de filtre  
    Set IP = CreateObject("WIA.ImageProcess")
```

```
'boucle sur l'ensemble des filtres disponibles  
For x = 1 To IP.FilterInfos.Count  
    Debug.Print "--> " & IP.FilterInfos(x).Name &  
    ""  
    Debug.Print IP.FilterInfos(x).Description  
    Debug.Print vbCrLf & "*****" & vbCrLf &  
    vbCrLf  
Next x  
End Sub
```

Remarque:

Les descriptions sont en anglais.

Les sous chapitres suivants sont la traduction du résultat de la macro "listerInformationsGeneralesFiltres".

N'hésitez pas à m'avertir si j'ai fait des erreurs dans la traduction...;o)

2.1.1.1. RotateFlip / Rotations et Retournements

Permet d'effectuer des rotations (par 90 degrés), ainsi que des retournements horizontaux et verticaux.

RotationAngle -

Propriété pour effectuer une rotation de 90, 180, ou 270 degrés. La valeur par défaut est 0.

FlipHorizontal -

Propriété pour appliquer un retournement horizontal si la valeur True est attribuée. La valeur par défaut est False.

FlipVertical -

Propriété pour appliquer un retournement vertical si la valeur True est attribuée. La valeur par défaut est False.

FrameIndex -

Propriété pour indiquer l'index d'un Frame, dans le cas où vous souhaitez modifier un Frame autre que le Frame actif. La valeur par défaut est 0.

2.1.1.2. Crop / Couper

Couper / Rogner une image en fonction des valeurs indiquées.

Left -

Définit la position à partir du bord gauche de l'image pour la coupe (valeur en pixels). La valeur par défaut est 0.

Top -

Définit la position à partir du bord supérieur de l'image pour la coupe (valeur en pixels). La valeur par défaut est 0.

Right -

Définit la position à partir du bord droit de l'image pour la coupe (valeur en pixels). La valeur par défaut est 0.

Bottom -

Définit la position à partir du bord inférieur de l'image pour la coupe (valeur en pixels). La valeur par défaut est 0.

FrameIndex -

Propriété pour indiquer l'index d'un Frame, dans le cas où vous souhaiteriez modifier un Frame autre que le Frame actif. La valeur par défaut est 0.

2.1.1.3. Scale / Redimensionner

Redimensionne une image en indiquant les hauteur et largeur maximum, tout en préservant les proportions si nécessaire.

MaximumWidth -

Définit la largeur maxi de l'image (en pixels).

MaximumHeight -

Définit la hauteur maxi de l'image (en pixels).

PreserveAspectRatio -

Appliquez la valeur True (Valeur par défaut) pour conserver les proportions.

Appliquez la valeur False pour que l'image s'étire en fonction des valeurs MaximumWidth et MaximumHeight.

FrameIndex -

Propriété pour indiquer l'index d'un Frame, dans le cas où vous souhaiteriez modifier un Frame autre que le Frame actif. La valeur par défaut est 0.

2.1.1.4. Stamp / Fusionner

Fusionne 2 images à partir des coordonnées horizontales et verticales spécifiées.

ImageFile -

Définit l'image que vous souhaitez fusionner.

Left -

Définit la position horizontale par rapport au bord gauche de l'image (en pixels). La valeur par défaut est 0.

Top -

Définit la position verticale par rapport au bord supérieur de l'image (en pixels). La valeur par défaut est 0.

FrameIndex -

Propriété pour indiquer l'index d'un Frame, dans le cas où vous souhaiteriez modifier un Frame autre que le Frame actif. La valeur par défaut est 0.

2.1.1.4. Exif

Ajoute ou supprime la propriété EXIF spécifiée.

Remove -

Indiquez la valeur True si vous souhaitez supprimer la propriété EXIF spécifiée. La valeur par défaut (False) permet d'ajouter la propriété EXIF spécifiée

ID -

Indique l'ID de la propriété que vous souhaitez ajouter ou supprimer.

Type -

Indique le type de valeur pour la propriété Exif que vous souhaitez ajouter (ignoré lors de l'utilisation la propriété Remove)

Value -

Indique la valeur de la donnée EXIF que vous souhaitez ajouter (ignoré lors de l'utilisation la propriété Remove)

FrameIndex -

Propriété pour indiquer l'index d'un Frame, dans le cas où vous souhaiteriez modifier un Frame autre que le Frame actif. La valeur par défaut est 0.

2.1.1.5. Frame

Ajoute ou supprime le Frame spécifié.

Remove -

Indiquez la valeur True pour supprimer le Frame spécifié (par son index). La valeur par défaut (False) permet d'insérer une image avant le frame spécifié.

ImageFile -

Indique l'objet image du Frame actif que vous souhaitez ajouter. (ignoré lors de l'utilisation la propriété Remove)

FrameIndex -

Pour les suppressions, indiquez l'index du Frame à supprimer. Pour les ajouts, le frame sera inséré avant l'index spécifié. La valeur par défaut est 0.

2.1.1.6. ARGB

Modifie les bits d'une image en fonction des données spécifiées.

ARGBData -

La propriété ARGBData Renvoie un vecteur de valeurs Long qui représente les données ARGB ([canal Alpha](#) & [couleurs RGB](#)) du Frame spécifié.

FrameIndex -

Propriété pour indiquer l'index d'un Frame, dans le cas où vous souhaiteriez modifier un Frame autre que le Frame actif. La valeur par défaut est 0.

2.1.1.7. Convertir

Convertit l'image dans un type spécifié.

FormatID -

Indiquez le format souhaité pour la conversion. Les constantes disponibles sont: `wiaFormatBMP`, `wiaFormatPNG`, `wiaFormatGIF`, `wiaFormatJPEG`, ou `wiaFormatTIFF`.

Quality -

pour les fichiers JPEG, appliquez une valeur entre 1 to 100 pour spécifier la qualité de compression. La valeur par défaut = 100.

Compression -

Pour un fichier TIFF, indiquez les formats : CCITT3, CCITT4, RLE ou Uncompressed pour spécifier le mode de compression. La valeur par défaut est LZW.

2.1.2. Comment utiliser les filtres

Les exemples suivants sont en partie adaptés de [l'aide MSDN](#):

2.1.3. Redimensionner une image

Vb

```
Sub redimensionnerImage()  
    Dim Img As WIA.ImageFile, IP As WIA.ImageProcess  
  
    'Création conteneur pour l'image à manipuler  
    Set Img = CreateObject("WIA.ImageFile")  
    'Création du gestionnaire de filtre  
    Set IP = CreateObject("WIA.ImageProcess")  
  
    'Chargement de l'image dans le conteneur  
    Img.LoadFile "C:\fourmiz.JPG"  
  
    'Ajoute le filtre pour redimensionner l'image  
    (Scale)  
    IP.Filters.Add IP.FilterInfos("Scale").FilterID  
    'Définit la largeur maxi pour le  
    redimensionnement  
    IP.Filters(1).Properties("MaximumWidth") = 90  
    'Définit la hauteur maxi pour le  
    redimensionnement  
    IP.Filters(1).Properties("MaximumHeight") = 90  
    'remarque :  
    'Les proportions sont conservées. Le filtre  
    prend en compte  
    'les ratios et adapte la taille pour ne pas  
    dépasser les valeurs maxi définies.  
  
    'Application du filtre à l'image  
    Set Img = IP.Apply(Img)  
    'Enregistre l'image redimensionnée  
    Img.SaveFile "C:\fourmizThumbnail.JPG"  
End Sub
```

2.1.4. Combiner deux images pour en créer une nouvelle

Cette procédure permet de créer une nouvelle image à partir de 2 images existantes.

Remarque:

Dans cet exemple, la 2eme image va s'incruster dans la première à partir de l'angle inférieur droit.

Les dimensions de la 2eme image doivent donc être plus petites que la première afin que celle-ci ne soit pas totalement masquée.

Vb

```
Sub combinerDeuxImages()  
    Dim Img1 As WIA.ImageFile, Img2 As WIA.ImageFile, IP  
    As WIA.ImageProcess  
  
    'Création conteneur pour l'image à manipuler  
    Set Img1 = CreateObject("WIA.ImageFile")  
    'remarque: vous devez créer un conteneur pour chaque  
    image  
    Set Img2 = CreateObject("WIA.ImageFile")  
    'Création du gestionnaire de filtre
```

```
    Set IP = CreateObject("WIA.ImageProcess")  
  
    'Chargement de la 1ere image dans le conteneur  
    Img1.LoadFile ("C:\fourmiz.JPG")  
  
    ' Remarque pour cet exemple :  
    'Les dimensions de la 2eme image doivent être plus  
    petites que la première afin  
    'que celle-ci ne soit pas totalement masquée  
  
    'Chargement de la 2eme image dans le conteneur  
    Img2.LoadFile ("C:\fourmizThumbnail.JPG")  
  
    'Ajoute le filtre pour combiner/fusionner  
    (Stamp)  
    IP.Filters.Add  
    (IP.FilterInfos("Stamp").FilterID)  
    '  
    IP.Filters(1).Properties("ImageFile") = Img2  
    'définit la position horizontale de la 2eme  
    image pour la fusion  
    IP.Filters(1).Properties("Left") = Img1.Width -  
    Img2.Width  
    'définit la position verticale de la 2eme pour  
    la fusion  
    IP.Filters(1).Properties("Top") = Img1.Height -  
    Img2.Height  
  
    'Application du filtre à la 1ere image  
    Set Img1 = IP.Apply(Img1)  
    'sauvegarde de la nouvelle image  
    Img1.SaveFile ("C:\fourmiz_ImageCombinee.JPG")  
End Sub
```

2.1.5. Fusionner Deux images verticalement

Cet exemple permet de créer une nouvelle image à partir de 2 images existantes en appliquant une jointure bout à bout verticalement.

Vb

```
Sub FusionVerticale_DeuxImages()  
    Dim Img1 As Object, Img2 As Object  
    Dim IP As ImageProcess  
    Dim Largeur As Long, Hauteur As Long  
    Dim V As Object, Img3 As Object  
    Dim C As Long  
    Dim i As Integer  
  
    'Création conteneur pour l'image à manipuler  
    Set Img1 = CreateObject("WIA.ImageFile")  
    'remarque: vous devez créer un conteneur pour chaque  
    image  
    Set Img2 = CreateObject("WIA.ImageFile")  
    'Création du gestionnaire de filtre  
    Set IP = CreateObject("WIA.ImageProcess")  
  
    'Chargement de la 1ere image dans le conteneur  
    '(l'image qui sera placée au dessus)  
    Img1.LoadFile ("C:\fourmiz.JPG")  
    'Chargement de la 2eme image dans le conteneur  
    '(l'image qui sera placée dessous)  
    Img2.LoadFile ("C:\fourmizThumbnail.JPG")  
  
    '-----ceation d'un support pour fusionner les 2  
    images  
    If Img1.Width > Img2.Width Then  
        Largeur = Img1.Width  
    Else  
        Largeur = Img2.Width
```

```

End If

Hauteur = Img1.Height + Img2.Height

C = &H80000004 'couleur de fond
Set V = CreateObject("WIA.Vector")

V.Add C
V.Add C
V.Add C
V.Add C

Set Img3 = V.ImageFile(2, 2)
Set IP = CreateObject("WIA.ImageProcess")
IP.Filters.Add IP.FilterInfos("Scale").FilterID
IP.Filters(1).Properties("MaximumWidth") =
Largeur
IP.Filters(1).Properties("MaximumHeight") =
Hauteur
IP.Filters(1).Properties("PreserveAspectRatio")
= False
Set Img3 = IP.Apply(Img3)
'-----

'reinitialisation des filtres
For i = 1 To IP.Filters.Count
IP.Filters.Remove i
Next i

'fusionne l'image1 dans le support
IP.Filters.Add (IP.FilterInfos("Stamp").FilterID)
IP.Filters(1).Properties("ImageFile") = Img1
IP.Filters(1).Properties("Left") = 0
IP.Filters(1).Properties("Top") = 0

Set Img3 = IP.Apply(Img3)

'fusionne l'image2 dans le support
IP.Filters(1).Properties("ImageFile") = Img2
IP.Filters(1).Properties("Left") = 0
IP.Filters(1).Properties("Top") = Img1.Height

IP.Filters.Add IP.FilterInfos("Convert").FilterID
IP.Filters(2).Properties("FormatID").Value =
wiaFormatJPEG
IP.Filters(2).Properties("Quality").Value = 90

Set Img3 = IP.Apply(Img3)

'sauvegarde de la nouvelle image
Img3.SaveFile ("C:\resultat_Fusion_Deux_images.jpg")
End Sub

```

2.1.6. Couper / Rogner une image

L'exemple suivant permet de couper une image afin de n'en récupérer que la partie supérieure gauche (Rapport 2/3).

Vb

```

Sub couperImage()
Dim Img1 As Object, IP As Object

'Création conteneur pour l'image à manipuler
Set Img1 = CreateObject("WIA.ImageFile")
'Création du gestionnaire de filtre
Set IP = CreateObject("WIA.ImageProcess")

'Chargement de l'image dans le conteneur
Img1.LoadFile ("C:\fourmiz.jpg")

```

```

'Ajoute le filtre pour Couper/Rogner l'image (Crop)
IP.Filters.Add (IP.FilterInfos("Crop").FilterID)

'La coupe sera effectuée à l'intérieur du cadre
défini ci dessous:
'*****
'définit la position à partir du bord gauche pour la
coupe
IP.Filters(1).Properties("Left") = 0
'définit la position à partir du bord supérieur pour
la coupe
IP.Filters(1).Properties("Top") = 0
'définit la position à partir du bord droit pour la
coupe
IP.Filters(1).Properties("Right") = Img1.Width / 3
'définit la position à partir du bord inférieur pour
la coupe
IP.Filters(1).Properties("Bottom") = Img1.Height / 3

'application du filtre
Set Img1 = IP.Apply(Img1)

'Sauvegarde de la nouvelle image
Img1.SaveFile ("C:\sauvegarde_Image_Coupee.jpg")
End Sub

```

2.1.7. Créer une image TIFF type multi-frame (multi-page)

Cet exemple Crée une image TIFF type multi-frame, à partir de 3 images existantes

Vb

```

Sub creationImage_Multipage()
Dim Img As ImageFile, Page2 As ImageFile, Page3 As
ImageFile
Dim IP As ImageProcess
Dim v As Vector

'Création des conteneurs pour les images à manipuler
Set Img = CreateObject("WIA.ImageFile")
Set Page2 = CreateObject("WIA.ImageFile")
Set Page3 = CreateObject("WIA.ImageFile")

'Création du gestionnaire de filtre
Set IP = CreateObject("WIA.ImageProcess")

'chargement des images
Img.LoadFile "C:\fourmiz.JPG"
Page2.LoadFile "C:\fourmiz_ImageCombinee.JPG"
Page3.LoadFile "C:\fourmizThumbnail.JPG"

'Ajoute un filtre pour gérer un premier
Frame/Multipage
IP.Filters.Add IP.FilterInfos("Frame").FilterID
'définit le contenu du Frame
Set
IP.Filters(IP.Filters.Count).Properties("ImageFile") =
Page2

'Ajoute un filtre pour gérer un deuxieme
Frame/Multipage
IP.Filters.Add IP.FilterInfos("Frame").FilterID
'définit le contenu du Frame
Set
IP.Filters(IP.Filters.Count).Properties("ImageFile") =
Page3

'Ajoute un filtre de conversion pour la nouvelle
image (au format TIFF)
'(TIFF est le seul format qui supporte la
sauvegarde en multi-frames)

```

```

IP.Filters.Add
IP.FilterInfos("Convert").FilterID
IP.Filters(IP.Filters.Count).Properties("FormatI
D") = wiaFormatTIFF

'Application du filtre à la 1ere image:
'(ajoute 2 frames dans la 1ere image, au format Tiff
Set Img = IP.Apply(Img)

'sauvegarde de la nouvelle image
Img.SaveFile "C:\monImageMultipage.tif"
End Sub

```

Remarque :
TIFF est le seul format qui supporte la sauvegarde en multi-frames.

2.2. Les propriétés des images

2.2.1. Lister les propriétés d'une image

La librairie WIA permet d'extraire toutes les propriétés des images:

Vb

```

Dim Img As ImageFile
Dim P As Property
Dim S As String

'Création conteneur pour l'image à manipuler
Set Img = CreateObject("WIA.imageFile")

'Chargement de l'image dans le conteneur
Img.LoadFile ("C:\fourmiz.JPG")

'Boucle sur la collection de propriétés
For Each P In Img.Properties
    S = P.Name & "(" & P.PropertyID & ") = "
    If P.IsVector Then
        S = S & " - vector data not emitted - "

    ElseIf P.Type = RationalImagePropertyType
Then
        S = S & P.Value.Numerator & "/" &
P.Value.Denominator

    ElseIf P.Type = StringImagePropertyType Then
        S = S & """" & P.Value & """"

    Else
        S = S & P.Value
    End If
    Debug.Print S
Next

```

2.2.2. Créer un nouveau TAG dans une image

La macro précédente montre comment lire les propriétés d'une image, mais il est aussi possible de les modifier.

Voici une procédure pour créer un nouveau TAG "Titre" dans une image.

Cet exemple utilise le filtre Exchangeable Image File (EXIF).

Vb

```

Sub creation_TAG_TITRE_copieImage()
    Dim Img As ImageFile
    Dim IP As ImageProcess
    Dim v As Vector
    Dim i As Integer

```

```

'Création conteneur pour l'image à manipuler
Set Img = CreateObject("WIA.imageFile")

'creation du gestionnaire de filtre
Set IP = CreateObject("WIA.imageProcess")

'création d'un vecteur
'(Un vecteur permet de créer une collection pour des
valeurs du même type.)
Set v = CreateObject("WIA.Vector")

'chargement de l'image
Img.LoadFile "C:\fourmiz.JPG"

'définit le filtre pour gérer l'EXIF
IP.Filters.Add IP.FilterInfos("Exif").FilterID
IP.Filters(1).Properties("ID") = 40091
    '40091 Titre
    '40092 Commentaire
    '40093 Auteur
    '40094 Mots clés

'spécifie le type de valeur pour le
propriété du filtre:
    'constante
VectorOfBytesImagePropertyType = 1101 (la valeur est un
vecteur)
    IP.Filters(1).Properties("Type") =
VectorOfBytesImagePropertyType
    'voir le fichier d'aide fourni avec la
librairie WIA pour
    'obtenir la liste des constantes
disponibles

'stocke une chaine de caractères dans le vecteur
v.SetFromString "Test de TAG 'TITRE' : utilisation
de WIA v2.0"

'applique au filtre le contenu du vecteur
IP.Filters(1).Properties("Value") = v

'application du filtre dans l'image
Set Img = IP.Apply(Img)

'sauvegarde de l'image
Img.SaveFile "C:\fourmiz_Test_EXIF.JPG"
End Sub

```

Ensuite allez dans l'explorateur Windows pour visualiser le résultat:

Faites un clic droit sur l'image "fourmiz_Test_EXIF.JPG"

Option Propriétés

Onglet Résumé (Description)

2.2.3. Vérifier si un fichier GIF est de type animé ou standard

La propriété IsAnimated permet de vérifier le format des images GIF.

Vb

```

Sub controleFormatImageGIF()
    Dim objImage As WIA.ImageFile

    'Création conteneur pour l'image à manipuler
    Set objImage = CreateObject("WIA.ImageFile")

    'chargement de l'image
    objImage.LoadFile
("C:\fichierImage.gif")

```

```

If objImage.IsAnimated Then
    MsgBox "Gif animé"
Else
    MsgBox "Gif standard"
End If
End Sub

```

3. Gérer les WebCams

3.1. Comment visualiser les images d'une WebCam dans un UserForm

Rien de plus simple:

Quand la librairie wiaaut.dll est installée sur votre poste, Cherchez les objets **DeviceManager Class** et **VideoPreview Class** dans la liste des contrôles supplémentaires. Insérez les 2 objets dans votre USF.

Et ensuite utilisez cette procédure:

Vb

```

Private Sub UserForm_Initialize()
    DeviceManager1.RegisterEvent
wiaEventDeviceConnected
    DeviceManager1.RegisterEvent
wiaEventDeviceDisconnected
End Sub

```

3.2. Capturer une image de la webCam

Cet exemple nécessite d'ajouter un contrôle Image nommé "Image1" et un bouton nommé "CommandButton1" dans l'UserForm. Le bouton permet de déclencher la capture qui va ensuite s'afficher dans le contrôle Image.

Vb

```
Option Explicit
```

```

Dim Di As DeviceInfo
Dim Dev As Device

Private Sub UserForm_Initialize()
    DeviceManager1.RegisterEvent wiaEventDeviceConnected
    DeviceManager1.RegisterEvent
wiaEventDeviceDisconnected

```

```

' (1) correspond au premier objet connecté
Set Di = DeviceManager1.DeviceInfos.Item(1)
' récupère la connexion active
Set Dev = Di.Connect

```

```

If Dev.Type = VideoDeviceType Then
    Set VideoPreview1.Device = Dev
End If
End Sub

```

```

Private Sub CommandButton1_Click()
    Dim Itm As Item
    Dim Img As ImageFile

```

```

' effectue la capture d'image
Set Itm = Dev.ExecuteCommand(wiaCommandTakePicture)

```

```

If Not Itm Is Nothing Then
    Set Img = Itm.Transfer

```

```

    If Not Img Is Nothing Then
        ' affiche l'image capturée dans un objet
        "Image"

```

```

        Set Image1.Picture =
        Img.FileData.Picture
    End If
End If

```

```

' pour enregistrer la capture sur le disque
' Img.saveFile "C:\monimageTest_WIA_V02.jpg"
End Sub

```

Retrouvez la suite de l'article de SilkyRoad et Bbil en ligne : [Lien85](#)

Vu dans la FAQ VB

Comment déterminer les variables d'environnement du système (Windows 2000 et +) ?

A partir de Windows 2000, vous pouvez retrouver facilement les variables d'environnement du système à l'aide de l'astuce suivante : (retourne dans la fenêtre d'exécution, toutes les variables d'environnement avec leur nom et leur valeur)

vb

```

Dim i As Integer
For i = 1 To 255
    If Environ(i) <> "" Then Debug.Print Environ(i)
Next

```

Dans un projet, vous pouvez utiliser, par exemple, le nom de la variable d'environnement dans votre code pour en extraire sa valeur :

vb

```
MsgBox Environ("USERNAME")
```

Retrouvez la FAQ VB en ligne : [Lien86](#)

Liens

Lien1 : <http://freed0.developpez.com/mysql/>
Lien2 : <http://dbprog.developpez.com/linux/hpagentsdebian/>
Lien3 : <http://sluce.developpez.com/faq/aix/>
Lien4 : http://blog.developpez.com/index.php?blog=49&title=construction_d_une_faq_aix&more=1&c=1&tb=1&pb=1
Lien5 : http://blog.developpez.com/index.php?blog=49&title=comment_se_connecter_en_ssh_sans_mot_de_passe&more=1&c=1&tb=1&pb=1
Lien6 : <http://gfx.developpez.com/tutoriel/java/strings/>
Lien7 : <http://dico.developpez.com/html/1710-Internet-Ajax-Javascript-Asynchrone-et-XML.php>
Lien8 : <http://www.gmail.com/>
Lien9 : <http://maps.google.com/>
Lien10 : <http://www.google.com/notebook/>
Lien11 : <http://dico.developpez.com/html/>
Lien12 : <http://code.google.com/webtoolkit/documentation/examples/desktopclone/demo.html>
Lien14 : <http://code.google.com/webtoolkit/terms.html>
Lien15 : <http://alexmorel.developpez.com/tutoriel/developpement-web/javascript/ajax/introduction/>
Lien16 : <http://www.developpez.net/forums/member.php?u=23681>
Lien17 : <http://code.google.com/webtoolkit/>
Lien18 : <http://moritan.developpez.com/tutoriels/java/gwt/presentation/>
Lien19 : <http://www.eclipsecon.org/summiteurope2006/>
Lien20 : http://blog.developpez.com/index.php?blog=12&title=eclipse_summit_europe_du_11_au_12_octobre&more=1&c=1&tb=1&pb=1
Lien21 : http://blog.developpez.com/index.php?blog=40&title=un_nouveau_livre_en_preparation_pour_net&more=1&c=1&tb=1&pb=1
Lien22 : <http://java.developpez.com/faq/maven/?page=utilisation#qr60>
Lien23 : <http://www.developpez.net/forums/profile.php?mode=viewprofile&u=27215>
Lien24 : <http://java.developpez.com/faq/maven/?page=documentation#qr46>
Lien25 : <http://java.developpez.com/faq/maven/?page=documentation#qr48>
Lien26 : <http://g-rossolini.developpez.com/tutoriels/php/les-formulaires-et-php5/?page=sommaire>
Lien27 : <http://www.parasoft.com/jsp/products/release.jsp?articleId=2355&type=Current>
Lien28 : http://blog.developpez.com/index.php?blog=92&title=communiqu_e_de_presse_parasoft_test_8_0&more=1&c=1&tb=1&pb=1
Lien29 : <http://webman.developpez.com/articles/dotnet/assemblysigning/>
Lien30 : http://blog.developpez.com/index.php?blog=9&title=event_mercredi_du_developpement_c_2_0_un&more=1&c=1&tb=1&pb=1
Lien31 : <http://mdeverdelhan.developpez.com/tutoriel/dynamiclight/tutoriel1/>
Lien32 : <http://mdeverdelhan.developpez.com/tutoriel/dynamiclight/tutoriel2/>
Lien33 : <http://c.developpez.com/faq/>
Lien34 : <http://xgql.free.fr/>
Lien35 : <http://xgql.developpez.com/xgql/presentation/>
Lien36 : <http://xml.developpez.com/faq/>
Lien37 : <http://dico.developpez.com/html/2069-Economie-administrateur-de-bases-de-donnees.php>
Lien38 : <http://dico.developpez.com/html/1239-Economie-developpeur.php>
Lien39 : <http://dico.developpez.com/html/137-Generalites-base-de-donnees.php>
Lien40 : <http://dico.developpez.com/html/510-Securite-confidentialite.php>
Lien41 : <http://dico.developpez.com/html/533-Securite-integrite.php>
Lien42 : <http://dico.developpez.com/html/2102-Securite-PCA-plan-de-continuite-dactivite.php>
Lien43 : <http://dico.developpez.com/html/2977-Securite-DoS-Denial-of-Service.php>
Lien44 : <http://dico.developpez.com/html/2976-Securite-disponibilite.php>
Lien45 : <http://dico.developpez.com/html/1236-Generalites-en-ligne-de-commande.php>
Lien46 : <http://dico.developpez.com/html/2104-Securite-log-file.php>
Lien47 : <http://dico.developpez.com/html/1239-Economie-developpeur.php>
Lien48 : <http://sql.developpez.com/sqlaz/dcl/>
Lien49 : <http://www.developpez.net/forums/profile.php?mode=viewprofile&u=3190>
Lien50 : <http://leoanderson.developpez.com/securisation-oracle/>
Lien51 : <http://www.developpez.net/forums/profile.php?mode=viewprofile&u=52427>
Lien52 : <http://dev.mysql.com/doc/refman/5.0/fr/account-management-sql.html>
Lien53 : <http://www.developpez.net/forums/member.php?u=53385>
Lien54 : <http://cyberzoide.developpez.com/securite/privileges-base-de-donnees/>
Lien55 : <http://www.oracle.com/technology/software/index.html>
Lien56 : <http://oracle.developpez.com/outils>
Lien57 : http://oraclestore.oracle.com/OA_HTML/ibeCZzpHome.jsp?minisite=10344&respid=1298576&grp=STORE&language=FR
Lien58 : <http://oracle.developpez.com/faq/?page=sommaire>
Lien59 : http://fr.wikipedia.org/wiki/Mozilla_Firefox
Lien60 : <http://extensions.geckozone.org/MenuEditor>
Lien61 : <http://adblock.mozdev.org/>
Lien62 : <http://bene.sitesled.com.nyud.net:8090/adblock.htm>
Lien63 : <http://www.pierceive.com/>
Lien64 : <http://extensions.geckozone.org/IETab>
Lien65 : <http://extensions.geckozone.org/GetletableMail>
Lien66 : <http://extensions.geckozone.org/TabMixPlus>
Lien67 : <http://extensions.geckozone.org/ScrapBook>
Lien68 : <http://joliclic.free.fr/mozilla/webdeveloper/>
Lien70 : <https://addons.mozilla.org/firefox/918/>
Lien71 : <http://extensions.geckozone.org/FireFTP>
Lien72 : <http://frenchmozilla.sourceforge.net/cz/>
Lien73 : <https://addons.mozilla.org/firefox/135/>
Lien74 : <https://addons.mozilla.org/firefox/655/>
Lien75 : <https://addons.mozilla.org/firefox/1843/>
Lien76 : <http://www.customizegoogle.com/>
Lien77 : <http://extensions.geckozone.org/Autohide>
Lien78 : http://ersplus.free.fr/spip/article.php?id_article=151
Lien79 : <http://extensions.geckozone.org/MenuEditor>
Lien80 : <http://extensions.geckozone.org/ExternalApplicationButtons>
Lien81 : <http://extensions.geckozone.org/BookmarksSynchronizer>
Lien82 : <http://www.geckozone.org/>
Lien83 : <http://windows.developpez.com/faq/vista/>
Lien84 : <http://www.microsoft.com/downloads/details.aspx?FamilyID=a332a77a-01b8-4de6-91c2-b7ea32537e29&DisplayLang=en>
Lien85 : <http://silkyroad.developpez.com/VBA/WindowsImageAcquisition/>
Lien86 : <http://vb.developpez.com/faq/>