



Developpez

Magazine

Edition de Avril-Mai 2007.

Numéro 9.

Magazine en ligne gratuit.

Diffusion de copies conformes à l'original autorisée.

Réalisation : Baptiste Wicht

Rédaction : la rédaction de Developpez

Contact : magazine@redaction-developpez.com

Index

Java	Page 2
Linux/BSD/Unix	Page 7
Développement Web	Page 10
DotNet	Page 15
C & C++	Page 20
XML	Page 25
Office	Page 28
Windows	Page 33
Delphi	Page 38
Ruby	Page 45
Conception	Page 51
2D/3D/Jeux	Page 56
Liens	Page 59

Editorial

Le nouveau magazine Developpez.com arrive avec les premières grandes chaleurs. Retrouvez comme d'habitude articles, livres, FAQ et autres ressources des différentes rubriques.

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

La rédaction

Comparatif Web

Découverte des moteurs de template PHP

Découvrez les principaux moteurs de template pour PHP ainsi que leurs avantages et inconvénients.

par **Guillaume Rossolini**

Page 10

Article Jeux

Présentation de la bibliothèque IrrKlang

Découvrez la nouvelle bibliothèque audio IrrKlang avec ce petit article qui vous montrera ses avantages et inconvénients

par **Pierre Schwartz**

Page 56



Les derniers tutoriels et articles

Découverte de WOJ pour Eclipse

WOJ est un ensemble de services qui permet à un développeur d'avoir accès à la documentation et aux sources de nombreux projets. Nous allons voir comment intégrer WOJ dans Eclipse.

1. Introduction

World of Java est un projet dont le concept est de donner un accès à la javadoc et aux sources d'un grand nombre de projets Java. Actuellement, la base de données de WOJ compte plus de 3600 révisions de projets. Les projets en questions sont nombreux et divers, on peut par exemple citer Jasper Reports, Hibernate, JFace, JDIC, SwingX ou encore IText.

La grande force de WOJ est son intégration dans les IDE, cela permet de gagner beaucoup de temps. En effet, plus besoin de rechercher la documentation d'un projet puisqu'on peut la trouver directement dans son IDE.

Dans cet article, nous allons parler du plugin pour Eclipse, mais sachez qu'il existe aussi une version pour Netbeans et une version pour IntelliJ Idea. WOJ fournit même un ensemble de tâches pour Ant.

WOJ est un programme payant, une version de démo est toutefois disponible pour un mois, passé ce délai, il vous faudra payer pour continuer à l'utiliser. Vous pouvez néanmoins obtenir une version gratuite pour les projets open source ou pour usage personnel.

2. Installation

Avant toute chose, il faudra bien sûr installer WOJ. Pour cela, rien de bien compliqué, il suffit de l'installer via l'update site.

Pour ce faire, allez dans le menu "Help", "software updates" et ensuite "Find and install". Sélectionnez ensuite la deuxième option ("Search for new features to install"). Ajoutez un nouveau site avec les infos suivantes :

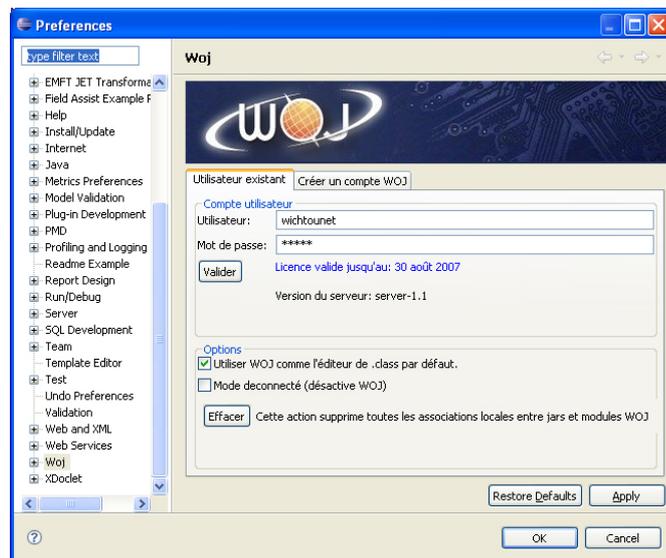
- Name : WOJ
- URL : <http://update.worldofjava.org>

Ensuite, il vous suffit de sélectionner ce site et de cliquer sur "finish". Choisissez ensuite le plugin WOJ qui devrait être arrivé dans la liste. A l'heure où j'écris cet article, la dernière version est la 1.1.2. Cliquez sur "next" et indiquez ensuite que vous acceptez les termes de la licence. Cliquez ensuite sur finish et attendez que l'installation soit terminée.

3. Configuration

Maintenant que vous avez installé WOJ, il va falloir le configurer. Normalement, la fenêtre des préférences de WOJ devrait s'afficher d'elle même au prochain lancement d'Eclipse, sinon, vous pouvez l'ouvrir depuis le menu WOJ en cliquant sur préférences.

Remplissez maintenant vos infos de connexion ou créez un nouveau compte en utilisant l'onglet créer un compte WOJ. Validez votre compte. La date de fin de votre licence va s'afficher sous vos identifiants de connexion.



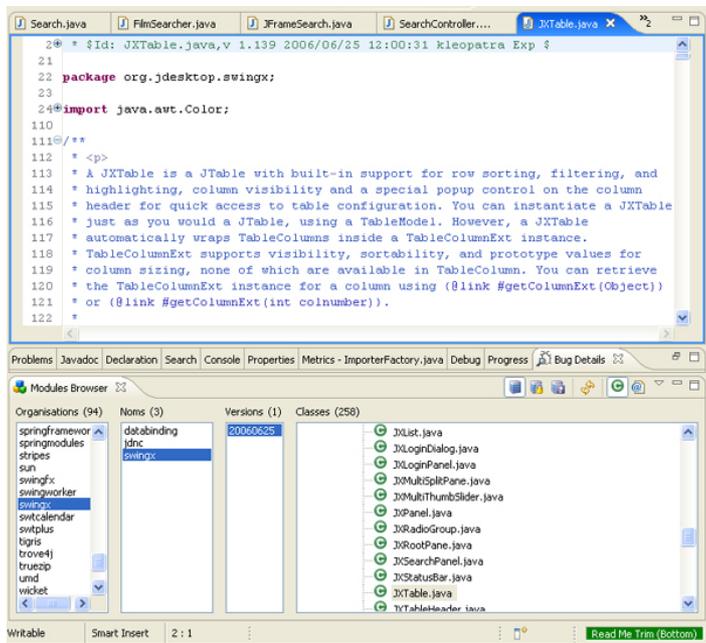
Si vous avez un proxy, il vous faut le configurer dans le sous menu de WOJ dans l'arbre à gauche.

Voilà, c'est tout ce que vous avez besoin de faire pour configurer WOJ, on peut maintenant passer à l'utilisation.

4. Utilisation

La première utilisation de WOJ est de pouvoir afficher la javadoc d'une classe soit standard soit d'un projet référencé par WOJ. Il suffit pour cela d'utiliser la combinaison Shift+F3 ou l'item correspondant du menu WOJ après avoir sélectionné une classe dans votre code. WOJ va ensuite ouvrir l'explorateur sur la javadoc de l'item en question. Vous pouvez bien sûr utiliser cela pour des constructeurs et des méthodes.

Une autre fonction de WOJ est de pouvoir visualiser les sources des divers projets de la base de données. Pour faire cela, aller dans le menu WOJ et choisissez l'explorateur de modules. Ensuite, dans la nouvelle vue qui est apparu, il vous faut choisir l'organisation puis le nom du projet que vous voulez visualiser. Et enfin choisissez la version dont vous désirez les sources. Il ne vous reste qu'à choisir la classe que vous désirez voir dans l'explorateur de package du projet. Double cliquez sur cette classe et vous en verrez le code source.



les sources, mais qui est dans WOJ, c'est WOJ qui vous affichera la source de la classe en question.

Vous pouvez aussi ajouter dans la base de données de WOJ des projets qui n'y sont pas encore. Si vous demandez à ouvrir la déclaration d'un projet qui ne se trouve pas dans la base de données WOJ, WOJ vous proposera de l'ajouter. D'abord, ce module sera ajouté à votre espace restreint et dès que l'équipe de WOJ aura validé le nouveau module, il sera visible de tous.

La dernière chose à savoir est que vous pouvez disposer d'un espace restreint avec la version professionnelle. En effet, vous pourrez ajouter dans un espace privé accessible que par vous et votre équipe, les sources et la documentation de votre projet.

5. Conclusion

En conclusion, je vous conseille d'essayer WOJ vous en serez à coup sûr séduit. Vous n'aurez plus besoin de rechercher les sources des projets que vous utilisez ni leur javadoc.

Je vous invite à consulter le site officiel de WOJ ([Lien1](#)) pour plus d'informations.

Retrouvez l'article en ligne de Baptiste Wicht : [Lien2](#)

Vous pouvez aussi faire cela directement dans un de vos projets qui utilise un librairie d'un des projets de WOJ. Si vous faites un Ctrl-clic pour visualiser la source d'une classe dont vous n'avez pas lié

Vu sur les Blogs

Forums : plugins Eclipse et applications RCP (Rich Client Platform)

Voici un nouveau forum ([Lien3](#)) dédié au développement de plugins et d'applications avec Eclipse RCP. Comme vous le savez tous, Eclipse est bien plus que l'EDI Java le plus populaire ([Lien4](#)), c'est avant tout une plateforme complète de développement.

Eclipse, grâce à son système de plugins, facilite grandement le développement de clients riches. Pour vous aider dans cette tâche, en plus des cours et tutoriels ([Lien5](#)) de developpez.com, des documentations officielles, vous pouvez désormais demander l'aide de la communauté dans le nouveau sous-forum Eclipse :

- **Forum d'entraide sur Eclipse RCP (Rich Client Platform) :** [Lien3](#)

Ce nouveau forum vient donc enrichir le forum Eclipse ([Lien6](#))

A ce propos, si vous voulez **vous aider** à enrichir et à animer toutes ces ressources sur la plateforme Eclipse, je vous invite à nous contacter à cette adresse : [eclipse\[AT\]redaction-developpez.com](mailto:eclipse[AT]redaction-developpez.com).

Retrouvez le billet de Ioan : [Lien7](#)

Java victime de l'antivirus

Depuis longtemps (la version 1) Java a la réputation d'être lent. La lenteur d'exécution s'est beaucoup améliorée pour ne plus être vraie avec les progrès de la JVM, notamment ceux du Garbage Collector et la compilation à la volée (JIT).

Il reste encore un principal reproche à Java que je partage, et qui est le temps de démarrage des applications. Qui n'a pas ralé devant les temps de démarrage des applications en Java, ou qui n'a pas vu ou participé à un troll sur les temps de lancement de la

JVM.

Je suis tombé sur cette discussion intéressante ([Lien8](#)) : l'auteur ainsi que plusieurs personnes dans les commentaires se plaignent de la lenteur des démarrages d'IDE ou des taches de type build. Beaucoup ont constaté que cela allait plus vite si l'antivirus (notamment McAfee) était désactivé.

J'ai fait les tests sur mon PC avec Norton Antivirus et effectivement les temps de démarrage de NetBeans ou Eclipse sont divisés par 3.

L'explication est que votre antivirus considère les fichiers jar comme des fichiers zip (ce qu'il sont en réalité) et donc l'intégralité de ces fichiers sont scannés lors des accès (lancement, étape de build...) ce qui ralentit énormément l'ordinateur. J'ai déjà vu un Pentium 4 avec 1 Go de RAM inutilisable pendant plusieurs minutes suite à l'ouverture d'un dossier contenant des ZIP que je gardais pour installer un environnement de développement.

La solution pour accélérer le démarrage de l'IDE est donc d'exclure de la surveillance de l'antivirus son répertoire d'installation et pour les builds il faut couper l'antivirus.

L'idéal serait de pouvoir exclure tous les .jar, mais on a déjà vu des virus en Java, enfin le terme est rapide puisqu'il s'agit d'applets exploitant des failles de la JVM de Microsoft ([Lien9](#)) (qui l'utilise encore???) et d'une applet qui téléchargeait un trojan Win32 ([Lien10](#)). Si vous avez des contre-exemples, n'hésitez pas.

PS : j'aime beaucoup un des commentaires qui dit que dans les ordinateurs dual-core, il y a un processeur pour l'antivirus et un pour notre utilisation....

Retrouvez le billet de ChristopheJ: [Lien11](#)



Les avantages du JUG

Vous avez sans doute vu passer l'information, mais la rubrique Java de developpez.com a le statut de virtual Java User Group ([Lien12](#)) ou VJUG.

Les JUG sont définis comme des groupes de gens qui se réunissent pour parler de Java. Les JUG à travers le monde vont du petit groupe d'une dizaine de personnes se retrouvant dans un bar à des groupes comme le BeJUG ([Lien13](#)) ou JUG Belge qui organise Javapolis et SpringOne ou encore le JUG de Brasilia : dfjug ([Lien14](#)) qui regroupe plus de 9000 membres.

Nous avons le statut de Virtual Java User Group car nous ne nous retrouvons pour l'instant qu'en virtuel sur Internet.

La communauté des JUG dispose d'avantages auprès de SUN et de différents organisateurs de conférences, parmi lesquels des réductions sur les tarifs d'entrées.

Vous pouvez vous aussi bénéficier de ces réductions en vous joignant au JUG et en nous en faisant la demande après avoir rejoint la communauté en suivant les instructions disponibles sur

le forum

La question est donc : mais quels sont ces avantages?

Il y en a trois actuellement en cours :

- le premier est pour JavaOne ([Lien15](#)), à San Francisco du 8 au 11 mai, avec une réduction de 100\$ sur votre inscription. A noter que le JUG qui réunira le plus de membres sur place disposera d'un entretien spécial avec un VIP (l'année dernière il s'agissait ni plus ni moins de James Gosling).

- le second est pour Jazoon 07 ([Lien16](#)) qui aura lieu à Zurich du 24 au 28 Juin, le JUG vous permettra d'avoir 30% de réduction sur votre inscription.

- le troisième est pour SpringOne ([Lien17](#)) du 20 au 22 juin à Anvers, avec une réduction de 10% (soit 75€)

Alors si vous comptez vous inscrire à une de ces conférences, n'hésitez pas à nous contacter pour profiter des réductions.

Retrouvez le billet de ChristopheJ : [Lien18](#)

Les livres Java

Les Cahiers du programmeur Swing

Ce cahier montre aux développeurs Java comment exploiter la très riche bibliothèque standard Swing à travers la création d'une application graphique 3D ergonomique et fiable.

Cet ouvrage décrit la création d'un logiciel libre d'aménagement d'intérieur développé avec Java 5 et Swing, de sa conception UML jusqu'à sa distribution, en passant par l'utilisation d'Eclipse, dont les astuces de productivité sont dévoilées. Le développeur y découvrira comment exploiter efficacement les nombreuses fonctionnalités de Swing : création d'arbres, de tableaux et de boîtes de dialogue, dessin dans un composant graphique, glisser-déposer, gestion d'opérations annulables, intégration d'un composant 3D, ...

Critique du livre par la rédaction (Vincent Brabant)

Tout d'abord, je dois vous signaler, avant que vous n'alliez plus loin dans la critique, que j'ai été relecteur technique de ce livre.

Mon avis aurait donc pu être faussé.

C'est la raison pour laquelle j'ai attendu que quelqu'un d'autre fasse une critique de ce livre, avant de publier la mienne :-).

Ce livre est le seul que j'ai lu jusqu'à ce jour qui explique aussi clairement comment développer une application Swing en utilisant le modèle MVC tout en faisant bien la distinction entre le modèle MVC de votre application et le modèle MVC (ou plutôt M-VC) des composants Swing.

Attention que ce livre n'est pas là pour vous faire découvrir tous les composants Swing qui sont à votre disposition, mais plutôt pour vous faire découvrir comment bien découper/structurer votre application Swing.

Malgré cela, on y apprend l'utilisation des JTable, JTree, ..., à afficher le même modèle sous des vues différentes (on visualise la même pièce en 3D, en 2D et sous forme de table), implémenter l'Undo/Redo, le Drag and Drop, l'internationalisation d'une application, ...

On y apprend de plus comment découper un projet, comment les personnes ont raisonné pour finalement arriver au design de cette application.

Tout au long de l'ouvrage, l'auteur compare également Swing et

AWT avec SWT et JFace, nous fait découvrir des modules Eclipse intéressants.

L'étude de cas que l'auteur a choisi pour ce livre est certainement pour beaucoup dans la qualité de l'ouvrage.

Dès le premier chapitre, on a envie d'en savoir plus, de connaître comment l'auteur s'y est pris pour réaliser son étude de cas.

Sachez également que l'auteur a aussi mis son étude de cas à la disposition de tous, sous licence open source, et qu'elle est totalement opérationnelle.

En conclusion, je ne peux que conseiller ce livre à tous ceux qui programment ou désirent programmer en Swing.

Ce livre me semble assez unique dans sa façon d'aborder Swing.

Pourquoi je ne donne que 4.5/5 à ce livre ? C'est la question que l'auteur m'a posée.

Je lui ai répondu que je mettrais 5/5 au livre qui m'expliquerait Swing à la façon "Head First/Tête première". Et le système des étoiles ne me permet pas de lui donner 4.9/5.

Retrouvez ce livre sur la rubrique Java : [Lien19](#)

EJB 3 Des concepts à l'écriture du code - Guide du développeur

Cet ouvrage est fondé sur les cours dispensés par le laboratoire SUPINFO des technologies Sun. Son objectif est de présenter et d'illustrer la nouveauté majeure de la dernière version de la plateforme Java Entreprise : EJB 3. Il a été conçu comme un guide de formation et un support de référence répondant aux questions concrètes de développement et de maintenance des services et des objets métiers. Il présente en complément une étude pratique complète basée sur un cas réel de développement d'une application entreprise.

Cet ouvrage conviendra aux développeurs Java désireux de s'initier aux systèmes de persistance des données, aux développeurs EJB 2 souhaitant évoluer vers un système plus modulable et plus rapide à mettre en place ainsi qu'aux développeurs J2EE cherchant à la fois un système performant de persistance de données et un guide pratique proposant de nombreux exemples de code prêt à l'emploi.

Critique du livre par la rédaction (Vincent Brabant)

Après nous avoir expliqué les couches et les modèles EJB dans le chapitre 1, on apprend au chapitre 2 le pourquoi de Java EE 5 et comment on est arrivé à EJB 3.

Je suis assez d'accord avec Alexis lorsqu'il dit dans sa préface qu'avec EJB 3, les EJB sont enfin matures. Même s'ils mériteraient encore une version 3.1 pour soulever des points d'ombre dans la spécifications EJB 3.

Les chapitre 3, 4 et 4 couvrent successivement les Sessions Beans, les Entity Beans et les Messages Driven Beans, parlant même un peu des Web Services dans les Sessions Beans. A chaque fois, on a droit à un bref rappel quand à leur utilisations en tant qu'EJB2, et ensuite, en tant qu'EJB3, avec l'usage des annotations, comme si c'était la seule façon d'écrire des EJB dans la spécification EJB3.

On a droit à un chapitre sur la nouvelle API JPA (Java Persistence API), un autre sur EJB-QL, et surtout les nouveautés introduites dans EJB3 pour ce QL. Ce langage semble enfin être mature et n'a plus trop à souffrir de ses limites qui existaient dans sa version 2. Puis encore un autre chapitre sur les transactions.

Les illustrations sont bien faites, et sont à propos. Le livre est agréable à lire.

Un bon livre sur les EJB3, même si on reste sur sa faim. En effet,

les chapitres concernant les Entity Beans, la persistance, et le EJB-QL auraient mérité d'être plus approfondis. Concernant les Sessions Beans et les Message Driven Beans, cela me semble assez complet. Mais les Entity Beans, Persistance et EJB-QL auraient vraiment mérités plus de place. J'aurai ainsi aimé voir plus d'informations quant à l'utilisation de l'EntityManager, dans un contexte hors Java EE, de savoir quelle est la meilleure classe de Collection à choisir (Un Set, une List, et pourquoi un Set plutôt qu'une List) et pourquoi il faut travailler avec les interfaces uniquement et pas avec les implémentations (ArrayList, HashSet), ...

Quelque chose qui m'a fait rigoler : ils indiquent que NetBeans aurait été créé par un Tcheque nommé Xelfi.

Il faut savoir que Xelfi était le nom du projet d'étudiants qui a donné naissance à NetBeans, car les étudiants voulaient faire un Delphi pour Java.

J'ai longtemps hésité pour la note entre 3.5 et 4.0. J'ai finalement décidé de lui donner 3.5. Un bon livre, bien illustré, mais on reste sur sa faim.

Retrouvez ce livre sur la rubrique Java : [Lien27](#)

Vu sur les FAQs

Comment écrire du texte dans une image ?

Version simple :

La méthode `drawString(java.lang.String,int,int)` de la classe `java.awt.Graphics2D` correspond parfaitement à ce que l'on cherche.

```
/** Ecriture dans une image 'img' */
Graphics2D g2d = (Graphics2D)img.getGraphics();
g2d.drawString("HelloWord",x,y);
```

Avec quelques raffinements :

On peut améliorer un peu le procédé et choisir la police et la couleur de notre texte. En fait, la classe `java.awt.Graphics2D` garde un certain nombre d'attributs qui sont appliqués à toutes les opérations de dessin ; la couleur principale, la couleur secondaire (de fond), la police, la déformation. Il suffit de changer ces valeurs en utilisant la méthode appropriée.

```
Graphics2D g2d = (Graphics2D)img.getGraphics();
g2d.setFont(new Font("Serif",Font.PLAIN,12));
g2d.setColor(Color.RED);
g2d.drawString("HelloWord",x,y);
```

A quoi correspond (x,y) ? :

Les coordonnées (x,y) que l'on donne à la méthode `drawString` correspondent au coin inférieur gauche de la première lettre du texte. Plus précisément, la valeur y correspond à la ligne 'de base' du texte. Pour les langues comme l'Arabe qui s'écrivent de droite à gauche, cette position correspond à la position du dernier caractère de la chaîne, donc de celui qui sera affiché le plus à gauche. (Si vous n'avez pas tout compris, faite le test. ;-))

Retrouvez la Q/R sur la FAQ Java GUI : [Lien20](#)

Comment utiliser le double-buffering software ?

L'affichage par défaut sur un composant `awt`, `swing`, ... de Java se

fait à l'aide d'un simple buffer (on voit l'image se tracer). Si on commence à faire des animations un peu complexes, il apparaît dès lors des problèmes de scintillements de l'image appelés clipping. Pour résoudre ce problème, on peut utiliser la méthode du double-buffering software (c'est une couche de la JVM qui s'en charge). Le double-buffering est le fait d'avoir un buffer où l'on dessine en mémoire et un buffer qui s'occupe de l'affichage. Les problèmes de rafraichissement, saut d'image et autres bugs visuels sont ainsi éliminés.

Voici un programme très simple se servant du double-buffering software.

```
public class DoubleBufferingSoftware extends Frame{
    // boucle d'affichage
    RenderingThread renderingThread = new
    RenderingThread();
    // buffer mémoire (2eme buffer)
    Graphics buffer;
    // image mémoire correspondante au buffer
    Image image;
    int x = 0; // coordonnée x de l'affichage du texte

    public DoubleBufferingSoftware(){
        //affichage
        setSize( 400, 400 );
        setVisible( true );

        // on démarre la bouche d'affichage
        renderingThread.start();
    }

    public void update(Graphics g){
        paint(g);
    }

    public void paint( Graphics g ){
        //création du buffer si il n'existe pas
        if(buffer==null){
```



```

        image = createImage(400,400);
        buffer = image.getGraphics();
    }
    //on dessine sur le buffer mémoire
    buffer.setColor( Color.white );
    buffer.fillRect( 0, 0, 400, 400 );
    buffer.setColor( Color.black );
    buffer.drawString( "affichage d'une ligne de
texte", x, 200 );
    x++;
    if(x>400) x = 0;
    // finalement, le buffer mémoire est dessiné dans
le buffer d'affichage
    g.drawImage(image, 0, 0, this);
}
class RenderingThread extends Thread {
/**
 * Ce thread appelle le rafraichissement de notre
fenêtre
 * toutes les 10 milli-secondes
 */
public void run(){
    while(true){
        try {
            repaint();
            sleep( 10 );
        } catch ( Exception e ) {}
    }
}
}
public static void main(String[] args){new
DoubleBufferingSoftware();}
}

```

Retrouvez la Q/R sur la FAQ Java GUI : [Lien21](#)

Nouveaux forums Graphisme

Je ne pouvais pas parler de ces questions de la partie graphisme de la FAQ GUI sans évoquer la création des nouveaux forums graphisme.

3 nouveaux forums sont en effet à votre disposition pour toutes vos questions :

- Graphisme : Forum d'entraide pour les APIs de dessin,

de graphisme, et multimédias. [Lien22](#)

- 2D : Forum d'entraide pour le dessin et le graphisme 2D (Java2D, JFreeChart, JGraph, JFreeReport, Chart2D, JChart2D, jCharts, JDrawing, ...). [Lien23](#)
- 3D : Forum d'entraide pour le dessin et le graphisme 3D (Java3D, JOGL, LWJGL, Xith3D, jMonkey Engine, Aviatrix3D, ...). [Lien24](#)

Qu'est ce que Maven ?

Maven est essentiellement un outil de gestion et de compréhension de projet.

- Maven offre des fonctionnalités de :
- Construction , compilation
- Documentation
- Rapport
- Gestion des dépendances
- Gestion des sources
- Mise à jour de projet
- Déploiement

Retrouvez la Q/R sur la FAQ Maven : [Lien25](#)

Qu'est ce que Continuum ?

Continuum est un serveur d'intégration continue principalement pour les projets Java.

Retrouvez d'autres infos sur le site officiel : [Lien26](#)

Continuum permet dont d'effectuer l'intégration d'un projet Java (compilation, tests, etc.) à des intervalles paramétrés.

Cette pratique s'accorde avec les techniques de développement "agiles" et Continuum est entièrement compatible avec les technologies d'intégration suivantes :

- Maven 1
- Maven 2
- Ant
- Shell scripts

Retrouvez la Q/R sur la FAQ Maven : [Lien27](#)



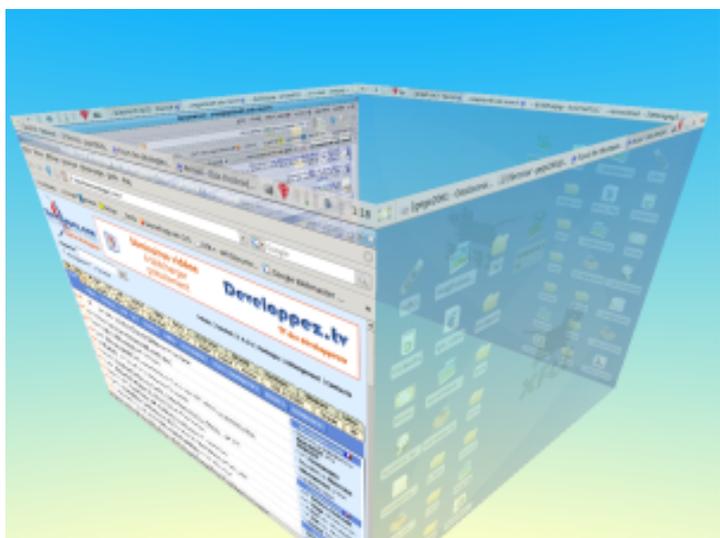
Les derniers tutoriels et articles

Installer beryl sous XFCE4

Le béryl, ou beryl, est un minéral composé de silicate de béryllium et d'aluminium, de formule $Be_3Al_2Si_6O_{18}$. Son nom vient du grec bérullos, cristal.

C'est wikipedia qui nous le dit ! C'est vrai mais ce n'est pas de minéraux que je vais vous parler aujourd'hui mais du gestionnaire de fenêtre, et plus précisément je vais vous expliquer comment l'installer sous Debian avec XFCE !

Alors Beryl ne vous dit peut être rien, mais cette copie d'écran vous est sûrement familière :



Eh oui Beryl est le bureau 3D de Linux. Enfin ce n'est pas le seul mais le plus connu et le plus avancé (il s'agit d'un fork de compiz).

Si vous ne possédez pas de carte NVidia, il faut installer aiglx qui est un module complémentaire pour le serveur X, il est à privilégier par rapport à Xgl qui est une sur-couche et par conséquent est plus gourmand en ressource.

Ayant une carte NVidia, je vais m'attarder uniquement sur cette partie.

j'ai commencé par installer les pilotes :

```
# apt-get install nvidia-glx
```

Ensuite, il faut modifier le fichier de configuration du serveur X. Vous pouvez utiliser la commande :

```
nvidia-xconfig
```

Ceci permet de remplacer le pilote nv par nvidia et dri par glx. En cas de problème lors du redémarrage du serveur, vous trouverez une copie de votre ancien fichier dans /etc/X11.

Voici à quoi doit ressembler la section **Module** :

```
Section "Module"
# Load "dri"
  Load "glx"
EndSection
```

Dans la section **Device**, ajoutez aussi les options **AddARGBGLXVisuals** et **XAANoOffscreenPixmaps** :

```
Section "Device"
  Identifier "NVIDIA Corporation NVIDIA Default Card"
# Driver "nv"
  Driver "nvidia"
  Option "AddARGBGLXVisuals" "On"
  Option "XAANoOffscreenPixmaps"
EndSection
```

Et ajoutez une section **Extensions** à la fin du fichier :

```
Section "Extensions"
  Option "Composite" "Enable"
EndSection
```

Redémarrez le serveur X et vous devriez apercevoir rapidement le logo NVidia au démarrage si tout c'est bien passé.

Ensuite il faut installer beryl (le gestionnaire de fenêtre) et emerald (le décorateur de fenêtre). Pourquoi emerald ? Bien sûr il existe plusieurs décorateurs mais ils sont basés sur des décorateurs existants :

- aquamarine : KDE
- heliodor : gnome
- yawd : non disponible dans les dépôts Debian
- compiz-gnome : le nom est clair

Étant sous XFCE, je ne souhaitai pas installer les dépenses liées à KDE ou gnome en plus emerald semble très populaire.

Revenons à notre installation. Il faut ajouter le dépôts beryl à votre /etc/apt/sources.list :

```
deb http://debian.beryl-project.org etch main
deb-src http://debian.beryl-project.org etch main

# wget http://debian.beryl-project.org/root@lupine.me.uk.gpg -O- | sudo apt-key add -
# apt-get update
```

Et c'est parti :

```
# apt-get install beryl emerald emerald-themes
```

N'oubliez pas les thèmes pour emerald, ils ne font pas partis des dépendances mais en leur absence, vous n'aurez pas de décoration à vos fenêtres.

Pour finir, le mieux est de lancer tout ce petit monde au démarrage de votre session XFCE. Pour cela créer un premier fichier pour beryl :

```
$ less ~/.config/autostart/beryl.desktop
[Desktop Entry]
Name=Beryl
Encoding=UTF-8
Version=1.0
Exec=beryl-manager
X-GNOME-Autostart-enabled=true
```

Et un second pour emerald :

```
$ less ~/.config/autostart/emerald.desktop
[Desktop Entry]
Name=Emerald
Encoding=UTF-8
Version=1.0
Exec=emerald --replace
```

Voilà, il ne vous reste plus qu'à faire le tour des options de Beryl

Enfin pour vous donner une idée, ma première installation m'a pris deux jours (beaucoup de redémarrage du serveur X, quelques problèmes pour lancer le décorateur de fenêtre, ...), mais je viens de refaire l'installation (spécialement pour vous) et j'ai mis 1 heure (j'avais gardé tous mes fichiers personnels) !

Il y aura plein de choses à dire sur le couple Beryl/emerald mais je préfère vous laisser les découvrir, au delà de l'apparence, certaines options deviennent vite très pratique ! Mais surtout n'oubliez pas que ces logiciels sont jeunes et il n'est pas rare de tomber sur un bug, par exemple quand je clique sur mon bureau, il se décale ou je ne peux pas utiliser le moteur de rendu zootreeves de emerald, cela le fait planter. Mais l'équipe de Beryl a pensé à tout, si cela vous énerve, faites un clic droit sur le beryl (qui je vous le rappelle est un minéral...) et vous pourrez choisir votre gestionnaire de fenêtre (pour revenir à xfwm4, par exemple).

Lire le billet de Gege2061 : [Lien28](#)

Les livres Linux

Le noyau Linux

Vous êtes-vous déjà demandé pourquoi Linux était aussi efficace ? Voulez-vous savoir si ses performances seront opérantes avec votre application préférée ? Avez-vous déjà jeté un œil au code source du noyau ? Souhaitez-vous simplement comprendre comment fonctionne un système d'exploitation moderne ? Si vous acquiescez à chacune de ces questions, alors cet ouvrage est fait pour vous. La lecture de cette troisième édition vous éclairera sur ce qui fait de Linux l'un des meilleurs systèmes et comment il fournit une réponse efficace au défi de l'ordonnement de processus, de l'accès aux fichiers et à la gestion de la mémoire dans une multiplicité d'environnements. La plupart des structures de données importantes, de nombreux algorithmes ou astuces de programmation en usage dans le noyau sont étudiés ; dans de nombreux cas, les fragments de codes pertinents sont analysés ligne par ligne. Par ailleurs, de nombreuses discussions relatives à Intel enrichissent cette nouvelle édition qui couvre les noyaux 2.6. Les auteurs introduisent chaque chapitre en expliquant l'importance et l'interaction entre le noyau et les utilitaires familiers des utilisateurs et des programmeurs. Cet ouvrage ne s'adresse pas exclusivement aux administrateurs système ou aux programmeurs, mais aussi aux étudiants et aux passionnés qui souhaitent comprendre comment fonctionnent réellement les choses à l'intérieur de la machine. Le noyau Linux est une visite guidée à travers des milliers de lignes de code : en route pour l'exploration !

Critique du livre par la rédaction (Mathieu Dalbin)

Ce livre a été écrit à partir, dans un premiers temps, de supports de cours et a été par la suite enrichi petit à petit par les deux auteurs en fonction des évolutions du noyau Linux. Cette évolution a permis d'aboutir à la troisième édition de cet ouvrage, qui traite désormais de la version 2.6 du noyau Linux. Ce livre s'adresse en priorité aux personnes souhaitant découvrir comment fonctionne le noyau Linux : ordonnancement des processus, gestion de la mémoire, communications entre processus, systèmes de fichiers... Cela dit, le livre ne s'adresse pas aux débutants, car de bonnes bases de programmation et des connaissances en

architecture des ordinateurs (x86 principalement) sont nécessaires pour aborder pleinement le contenu technique du livre.

En effet, on retrouve des pages d'explication sur le fonctionnement interne du noyau et une liste de détails techniques, qui apparaît être très exhaustive : rien n'est oublié ! Des bouts de codes en C et en assembleur, ainsi que des schémas explicatifs, agrémentent un texte qui se lit aisément.

Au sommaire de cette exploration du noyau, on retrouve donc les éléments classiques mais indispensables de tout noyau de système d'exploitation : gestion des processus et ordonnancement, gestion de la mémoire et pagination, communication entre processus, gestion des interruptions, synchronisation, modules, systèmes de fichiers, et bien d'autres chapitres. Ce livre semble indispensable pour quiconque s'intéressant à la structure du noyau Linux et voulant en modifier les sources. Ce livre peut également être utile aux programmeurs de modules ou de pilotes de périphériques par exemple, tant son exhaustivité est exemplaire. Un "Must" !

Retrouvez ce livre sur la rubrique Linux : [Lien29](#)

SUSE Linux 10.1

La migration ou l'installation d'un poste de travail sous Linux représente aujourd'hui une alternative fiable à l'utilisation de Windows. SUSE Linux 10 de l'éditeur Novell est la distribution la mieux adaptée à cet exercice tant sa prise en main est rapide et son intégration parfaite (son outil de configuration YaST, mondialement renommé, est à la base de son succès et garantit un passage en douceur de Windows vers Linux). Loin des clichés habituels liés à Linux (complexe, réservé aux professionnels...) ce livre présente aux habitués de Windows et aux débutants sous Linux comment migrer simplement. Il détaille dans un style très accessible et agrémenté de nombreux exemples et captures d'écrans toutes les notions élémentaires, les étapes d'installation, de configuration, d'administration et de dépannage. Afin que les habitués de Windows retrouvent tous leurs repères (et leur productivité) l'auteur présente les logiciels les plus adaptés pour une utilisation professionnelle comme personnelle : l'environnement graphique KDE, les suites bureautiques, tous les

logiciels pour Internet, les outils multimédias pour lire, convertir et créer tous les formats audio et vidéo...

Critique du livre par la rédaction (S.K. Martin)

Fiat Linux ! Et tout un nouvel univers s'ouvre.

Né en 1964, tout comme Unix, j'avais plusieurs fois au cours de mon existence entendu parler de LINUX, mais comme une grande majorité cela restait le côté obscur de la force...

Ayant par deux fois tenté seul d'aborder la bête au travers d'un liveCD Knoppix (CD auto amorçant comprenant une installation complète de Linux), ces deux tentatives peuvent se résumer par deux échecs. Rétrospectivement, je peux dire aujourd'hui que je n'y avais rien compris.

Puis il y a peu un membre de developpez m'a à nouveau convaincu de plonger du côté obscur avec SUSE 10.2. Après quelques galères de partitionnement et trois installations, cette opération s'avéra finalement salutaire car mon autre système, sans doute jaloux, vira à l'écran bleu ...

Voici où ce livre intervient ! Dès l'introduction, un électro-choc, une prise de conscience: Il existe autre chose, et pas des moindres !

Une fois les bases de la philosophie de Linux posées, l'auteur vous prend par la main étape par étape pour télécharger, installer puis

configurer SUSE Linux. Chaque étape étant parfaitement décrite et illustrée de copies d'écrans. Il n'est pas possible de vous perdre en route, tous les détails qui pourraient vous déstabiliser par rapport à votre système habituel sont répertoriés et expliqués (différences de comportement du clavier, de la souris, des installations de logiciels...).

Votre système installé, il est temps de passer à la pratique, en douceur d'abord avec la découverte de l'environnement graphique KDE (rassurant lorsque l'on vient de Windows), la configuration du système avec Yast, et, l'utilisation des applications sous Linux (travail ou détente). Puis un peu plus hardu, pour les intoxiqués dont je fais partie, le mode console. Mais encore une fois des explications claires et détaillées mettent l'interpréteur de commande à la portée du néophyte que je suis !

Pour finir un chapitre dédié aux dépannages

En conclusion, n'étant que débutant en la matière je ne puis me permettre de juger le fond technique de ce livre, ma critique se limitera donc à mon expérience propre. J'ai trouvé dans cet ouvrage un accompagnement parfait pour aborder Linux en toute sérénité. En bref comment un jeune padawan peut il dompter le côté obscur de la force en quelques chapitres et sans crainte. Ce livre est non seulement un excellent guide dans l'installation de SUSE Linux, mais aussi une bible à garder sous le coude pour l'utilisation courante de SUSE Linux

Retrouvez ce livre sur la rubrique Linux : [Lien30](#)

Découverte des principaux moteurs de template en PHP

1. Préambule

In the end you will always have an intersection point between presentation and logic. There is no way around it since the final thing you present is a merge of the two. The whole question of templating is about where you put this intersection point.

- Rasmus Lerdorf (créateur de PHP) dans la mailing list principale de PHP (<http://www.php.net/>).

C'est exactement ce qui transparaîtra de ce tutoriel : certains moteurs ont des gabarits passifs, tandis que d'autres déportent une grande partie de la logique applicative dans le fichier de gabarit.

Nous verrons les avantages et les inconvénients des deux méthodes (laisser le script PHP en charge de la logique ou bien la déporter dans le gabarit) au travers d'une exploration sommaire des principaux moteurs du marché.

GrandFather a porté mon attention sur une distinction faite couramment par les gens : un système de templates désigne les scripts construits selon l'idée de Rasmus (en PHP mais avec peu de code dans les fichiers HTML, il suffit donc d'exécuter le code qui peut être autonome), tandis que le moteur de templates met en place une méthode fondée sur une syntaxe spécifique (non PHP) qui doit donc être interprété par une classe PHP. Je me concentrerai ici sur cette seconde catégorie.

Cet article n'est pas un tutoriel complet sur chaque moteur. Au contraire, ce n'est qu'un survol de ce qu'il est possible de faire avec chacun d'eux en vue d'atteindre un objectif précis.

2. ModeliXe

ModeliXe est un projet français qui a été transmis à Florian Perrichot. La liste des contributeurs est longue et j'y ai retrouvé des noms que je connaissais : Sébastien Hordeaux (créateur de Waterpoo/PHPEdit) et Frédéric Bouchery (ancien blogueur aux billets intéressants).

Aujourd'hui, l'ancienne page existe mais elle n'est plus maintenue. Le site officiel, quoi qu'en dise Google.com, est désormais hébergé sur le site personnel de Florian ([Lien31](#)).

2.1. Mon avis

2.1.1. Avantages

De nombreuses personnes connues ont participé à l'élaboration de ModeliXe, ce qui donne un a priori favorable. Il a une gestion du cache.

2.1.2. Inconvénients

La configuration de ModeliXe repose sur des constantes du scope global plutôt que sur des constantes de classes.

Le gabarit en XML est trompeur, dans la mesure où des variables

doivent parfois être insérées dans le document à l'intérieur des propriétés des balises du gabarit : pourquoi écrire ces variables avec une syntaxe XML à cet endroit ?

Syntaxe XML douteuse :

```
<meta name="generator" content="<mx:text id="meta_generator"/>" />
```

2.1.3. Conclusion

ModeliXe me semble peu intéressant, dans la mesure où il n'apporte pas de nouveauté par rapport aux autres solutions, le support est à l'abandon et la communauté est inexistante.

3. phpBB2 (PHP Base Library - phplib)

PHP Base Library ([Lien32](#)) est l'une des premières bibliothèques permettant d'utiliser un système de gabarits.

L'idée de la phplib est d'utiliser des fichiers de gabarit (l'extension classique est ".ihtml") qui contiennent le code HTML utilisé. Ces gabarits sont de véritables patrons, ils définissent uniquement une structure ou des éléments de structure. C'est un moule qui ne comporte aucun élément de logique.

PhpBB2 ([Lien33](#)) est un script de forum très populaire.

Il y a de cela plusieurs années, le phpBB Group a effectué un très gros travail autour du template de la PHPLib afin de le rendre plus performant. C'est la version phpBB2 (dont le script de forum sont actuellement en bêta) que je vais utiliser ici, plutôt que la version PHPLib qui manque de maturité. L'extension de fichier est ".tpl".

3.1. Mon avis

3.1.1. Avantages

Ce moteur utilise PHP et les commentaires HTML, ce qui le rend très facile d'accès (car le développeur et le graphiste n'ont pas besoin d'apprendre un nouveau langage esotérique).

Ce moteur permet de faire la distinction entre la logique applicative et la couche de présentation.

3.1.2. Inconvénients

PHP < 5

Les messages d'erreurs pourraient être plus explicites.

3.1.3. Conclusion

J'aime beaucoup cette manière d'utiliser les templates. C'est simple à utiliser et l'idée des commentaires HTML est excellente, car il n'y a pas besoin d'apprendre une nouvelle syntaxe.

4. Smarty

Smarty est un projet du framework PEAR, qui est connu pour sa formidable quantité d'extensions et de classes PHP.

L'idée est de déporter la logique d'affichage depuis le script PHP vers le gabarit. Smarty se charge de tout ce qui se rapproche de près ou de loin de l'affichage du contenu dans la page finale, ce qui inclut certains détails de programmation comme les boucles (par exemple pour parcourir les tableaux).

Smarty ([Lien34](#))

4.1. Mon avis

4.1.1 Avantages

La classe fondamentale est simple à réutiliser. Il suffit de placer le répertoire de Smarty dans un dossier accessible par tous les utilisateurs d'un même serveur, puis de dériver la classe Smarty pour chaque site Web hébergé sur ce serveur.

Le code PHP est épuré de toute présentation. Cela le rend bien plus lisible, c'est indéniable.

Il est évident que Smarty est prévu pour utiliser une mise en cache.

Ce moteur permet de faire la distinction entre la logique applicative et la logique de présentation.

4.1.2 Inconvénients

PHP < 5

Pour être efficace, il faut apprendre à utiliser le langage de script de Smarty.

Le code du gabarit contient à la fois du HTML et une certaine logique applicative (celle liée à la présentation).

4.1.3 Conclusion

Une fois habitué à la syntaxe de Smarty, le développement va vite. Je me demande encore pourquoi avoir déplacé dans le gabarit ce qui ressemble à de la logique applicative (les boucles) mais cela a son utilité, quoique je n'en parle pas ici.

L'objectif principal est atteint : la logique applicative (récupération et traitement des informations) est séparée de la logique de présentation (affichage des informations). Le souci que je vois est que, pour y parvenir, Smarty est obligé d'utiliser du code dans les fichiers de gabarit. À mon sens, un gabarit est quelque chose de complètement passif, un outil utilisé par un autre outil, un simple moule. Avec Smarty, ce moule est actif.

Smarty dispose de plugins intéressants mais il pourrait aller bien plus loin. Par exemple, le plugin de sélection de date pourrait (devrait) produire une boîte JavaScript/AJAX sous forme d'un calendrier : ce serait une solution bien plus ergonomique que les trois <select> actuels. Je ne doute pas qu'un tel plugin ait été développé par un tiers : je ne parle pas ici des contributions qui ne sont pas intégrées dans la classe principale.

Ce qui me gêne avec Smarty est la forte similitude entre le code de **void/home.php** et celui de **smarty/home.php** (cf. plus haut). Certes, le code lié à la présentation des données n'est plus dans un script PHP... Mais y a-t-il une réelle différence ? Le graphiste de l'équipe acceptera-t-il d'apprendre la syntaxe de Smarty pour composer les interfaces ?

Attention : Il faut absolument préparer toutes les informations dans des tableaux associatifs avant de commencer à appeler la

méthode **assign()**, sans quoi l'imbrication de boucles avec Smarty peut être un problème. Ce n'est pas un problème en soi, il faut juste y penser.

5. TinyButStrong

TinyButStrong ([Lien35](#))

L'idée avec TinyButStrong est de réutiliser autant que possible tout le code existant, que ce soit du PHP ou du HTML.

L'auteur, Skrol29 ([Lien36](#)), est quelqu'un de très ouvert et toujours disponible pour les utilisateurs de TBS. Je ne saurais trop le remercier pour sa patience après sa lecture de la première version de ce comparatif. C'est toujours un plaisir de voir qu'une communauté conserve son personnage moteur.

Skrol29 a minutieusement révisé toute la partie "TinyButStrong" de ce comparatif. J'ai conservé certains de mes commentaires mais le côté technique est corrigé. Il ne subsiste pas d'erreurs.

5.1. Mon avis

5.1.1. Avantages

PHP 4 et 5.

Possibilité d'installer et de développer des plugins.

La syntaxe "relative" du gabarit peut sembler déroutante, mais en réalité elle est ingénieuse : elle permet de créer des gabarits XML depuis l'intérieur d'une application comme Office2007 ou OpenOffice. D'ailleurs, un plugin OpenOffice existe pour TBS...

TBS est prévu pour utiliser des gabarits 100% compatibles W3C, ce qui permet à un graphiste de créer les documents avec son outil favori.

5.1.2. Inconvénients

Cela relève du commentaire personnel mais je n'adhère pas toujours à la manière de coder, notamment l'utilisation de constantes dans le scope global plutôt qu'à l'intérieur d'une classe (groupement de classes dans un même script, un même paramètre ou une même méthode peut avoir diverses utilisations).

Construire des requêtes SQL n'est pas la vocation de TBS (et j'adhère à ce refus) mais la classe permet néanmoins d'exécuter des requêtes. Cependant, aucune protection n'est appliquée aux requêtes SQL envoyées par MergeBlock() ; il appartient au développeur de le faire avec la bonne fonction : par exemple pour MySQL, il s'agit de `mysql_real_escape_string()`.

5.1.3. Conclusion

TBS est le moteur de gabarits qu'il m'a été le plus difficile de maîtriser. J'ai finalement réussi à atteindre mon objectif (reproduire le document initial) : je le dois à l'aide patiente de son auteur, Skrol29.

6. Virtual Template

VTemplate est un projet français.

Le principe est le même que pour phplib, à savoir utiliser des commentaires HTML dans le gabarit afin de construire un moule de document.

6.1. Mon avis

6.1.2. Inconvénients

J'ai cru lire que VTemplate 1.3.3 était écrit en PHP 5, j'ai dû faire erreur. Il s'agit plutôt d'une classe PHP 4 compatible PHP 5 (comme toutes les classes PHP 4, en fait). Je préfère ne pas savoir quelles modifications ont été apportées pour valider cette compatibilité...

Les méthodes retournent -1 en cas d'erreur. Ce n'est pas une valeur assimilable à FALSE... Impossible d'utiliser la négation `if(!open())` ou la comparaison booléenne `if(open() === FALSE)`. De plus, aucune constante n'est définie avec la valeur -1. Par conséquent, la sémantique de la gestion d'erreurs est inexistante, ce qui rend le code difficile à lire.

Le changement entre la version "sans mise en cache" et la version "avec mise en cache" est complexe et peu flexible : il faut télécharger des scripts différents mais au nom identique depuis le site officiel. Il est possible d'utiliser en permanence la version avec gestion du cache mais au prix d'un numéro de version (1.3.0 au lieu de 1.3.3 pour la version sans mise en cache).

Les classes utilisées par VTemplate ont des noms génériques ("Err" et "Session"), ce qui a de fortes chances de poser des collisions lors de l'intégration au projet.

La configuration se fait au moyen de constantes situées dans le scope global. Il est notamment possible de modifier le tag d'ouverture des variables du gabarit mais pas le tag de fermeture. Des noms de constantes comme "VTP_ALL" et "VTP_CACHED" auraient été plus heureux et il aurait été encore plus adapté de les placer à l'intérieur d'une classe écrite en PHP 5, ce qui donnerait des noms comme "VTemplate::ALL" et "VTemplate::CACHED". Enfin, le répertoire des fichiers de cache n'est pas configurable par gabarit.

6.1.3. Conclusion

Il m'a été difficile d'utiliser VTemplate car je m'attendais à une classe prête à l'utilisation, or j'ai été contraint de lire son code pour la déboguer. Les bugs sont nombreux, c'est assez déplaisant.

Je n'ai généralement ni l'envie ni le temps de déboguer une classe avant de pouvoir l'utiliser, VTemplate n'a donc pas ma préférence.

Si quelqu'un a vécu une expérience différente, qu'il n'hésite pas à me le faire savoir. C'est surtout dommage pour la gestion du cache car, au vu de la rapidité du moteur selon mes tests, je m'attendais à quelque chose d'impressionnant.

7. Conclusion

7.1. Comparaison

Mes conclusions peuvent être subjectives, à vous de vous faire une idée.

7.1.1. Les bibliothèques

	Version comparée	Qualité du code	Syntaxe du gabarit
Smarty	2.6.14	recommandations PEAR	recommandations PEAR

phpBB2	1.10.2.5	ok + commentaires	commentaires HTML
TinyButStrong	3.2.0	homogène	configurable
VTemplate	1.3.3	médiocre	peu configurable
ModeliXe	1.2.2	moyenne	XML ou PEAR

	Cache	PHP	Licence
Smarty	Oui	4	LGPL
phpBB2	Aucun	3	GPL
TinyButStrong	Oui	4 adapté 5	LGPL
VTemplate	En théorie	4	GPL
ModeliXe	Oui	4 compatible 5	LGPL

PEAR a défini des conventions de codage ([Lien37](#)).

Informations sur la license GPL : site officiel ([Lien38](#)), Wikipedia ([Lien39](#)).

Informations sur la license LGPL : site officiel ([Lien40](#)), wikipedia ([Lien41](#)).

7.1.1. Tests de performances

J'ai entièrement refait les tests de performances des scripts depuis que j'ai publié ce comparatif pour la première fois.

Je vais décrire ici en détail ma méthode et fournir le script utiliser. Si vous obtenez des résultats différents, je vous serais reconnaissant de me les faire parvenir. De même, si ma méthode ne vous semble pas appropriée, merci de me le faire savoir.

J'utilise l'utilitaire Apache Bench (ab.exe), fourni en standard avec ma distribution d'Apache 2.0.59 et je tourne sous Windows.

Je doute que ma configuration système soit très utile mais la voici tout de même :

- Système : Windows XP SP2 ;
- Processeur : Intel Pentium M cadencé à 1300 MHz ;
- RAM : 512 Mo ;
- Apache : 2.0.59 ;
- PHP : 5.2.1 (Zend Core 2.0).

Skrol 29 (TinyButStrong) m'ayant convaincu de l'inutilité de ma méthode antérieure, j'ai adapté les scripts de ce comparatif pour éviter d'utiliser la base de données. Toutes les données sont codées en dur dans le script db-connect.php et les appels à MySQL sont supprimés.

Ma démarche est constituée des actions suivantes :

1. Supprimer les fichiers de log ;
2. Vider les répertoires des fichiers de cache et de compilation des différents moteurs ;
3. Récupérer des statistiques préliminaires (ab.exe) ;
4. Calculer la moyenne pour chaque moteur ;
5. Supprimer les quelques rapports erronés générés par Apache Bench ;
6. Calculer les statistiques finales (le comparatif).

Les statistiques sont récupérées sur 5000 chargements consécutifs par Apache Bench, pour chaque moteur. L'ensemble des tests est fait 30 fois (c'est-à-dire 30 fois 5000 chargements de chaque script) afin d'obtenir une moyenne fiable. La valeur utilisée dans le rapport est le nombre de requêtes par seconde.

À la fin de l'exécution du script de test, une fois les rapports erronés supprimés, nous calculons l'écart entre le nombre de requêtes par seconde du script de référence et le nombre de requêtes par seconde de chaque autre script : résultat = (référence / moteur) * 100. Le nombre obtenu est un pourcentage : par exemple, "150%" signifie que le moteur considéré est parvenu à exécuter 50% de chargements de moins par seconde, par rapport au script de référence. Une autre manière de le comprendre est : "ce moteur met 150% du temps du script de référence pour effectuer le même nombre de chargements".

Fichiers du test de performances : [Lien117](#)

7.1.2. Mon expérience avec ces bibliothèques (sans cache)

	Prise en main	Flexibilité	Débogage
Smarty	Facile	Immédiate	Facile
phpBB2	Facile	Rapide	Classique
TinyButStrong	Complexe	Immédiate	Complexe
VTemplate	Facile	Problématique	Classique
ModeliXe	Facile	Immédiate	Classique

Classement (décroissant) :

1. Smarty ;
2. ModeliXe ;
3. phpBB2 / VTemplate ;
4. TinyButStrong.

7.1.3. Leur gestion du cache

Voici un tableau récapitulatif de mon expérience avec ceux des moteurs de gabarits (de ma liste) qui proposent une gestion du cache.

	Adaptation au projet	Performances/Référence
Smarty	immédiate	116%
phpBB2	Non prévu pour	n/a
TinyButStrong	malaisée	113%
VTemplate	Après débogage	144%
ModeliXe	immédiate	117%(PEAR) 114%(XML)

Classement (décroissant) :

1. ModeliXe / Smarty / TinyButStrong ;
2. VTemplate.

7.2. Classements

Ces classements dépendent de mon interprétation des comparaisons ci-dessus : je ne saurais trop vous recommander de vous construire votre propre opinion !

Séparation entre la logique et la présentation (du plus au moins adapté) :

1. phpBB2, VTemplate et ModeliXe n'ont aucun élément de programmation au sein du gabarit, ils se contentent de définir des zones à remplir, à répéter ou à ignorer ;
2. Smarty et TinyButStrong optent pour des gabarits plus complexes afin de simplifier le code PHP.

Nous retrouvons ici les deux groupes évoqués en introduction : d'une part (en seconde position ici), les moteurs incorporant le point de fusion dans le gabarit et, d'autre part (en première position selon moi), ceux le laissant dans le code PHP.

Nous n'avons cependant pas de grand gagnant, il nous faut donc trouver un autre argument que la possibilité de séparation entre logique et présentation.

Simplicité de la syntaxe du gabarit (par ordre décroissant) :

1. phpBB2 : Des commentaires HTML comportant les mots BEGIN et END ;
2. VTemplate : Des commentaires HTML avec une touche de XML ;
3. ModeliXe : Une adaptation de la syntaxe XML ;
4. Smarty : Une nouvelle syntaxe fondée sur les accolades et comportant des éléments de programmation ;
5. TinyButStrong : Une nouvelle syntaxe fondée sur les crochets et comportant des éléments de programmation.

7.3. Épilogue

Je ne vous recommande pas la lecture de ce qui suit si vous n'avez pas étudié vous-même les différents moteurs.

Je rappelle ici l'objectif principal qui doit être atteint par un moteur de gabarits : permettre une séparation efficace du code source de présentation (par exemple HTML) et du code source de logique applicative (PHP).

Pour être honnête, un seul des cinq moteurs testés me semble convenable. Ce n'est bien sûr qu'un avis personnel et conditionné par ce que j'ai énoncé en introduction : chaque développeur devrait avoir son opinion sur ce sujet.

Smarty est le plus mature et le plus propre. Cependant, la syntaxe utilisée dans ses gabarits est tellement proche du langage PHP lui-même que la séparation logique/présentation est floue, voire inefficace. Je ne suis pas certain d'être d'accord avec le qualificatif de moteur de templates. Je ne dis pas que Smarty est inutile, mais plutôt qu'il n'est pas utile en tant que moteur de gabarits.

phpBB2 a le moteur le plus facile à utiliser et le plus rapide mais il ne gère pas la mise en cache. Ce n'est pas un défaut, c'est simplement un avantage qu'il n'a pas.

TinyButStrong promet de grandes choses mais a dû faire des concessions pour accéder aux requêtes de ses utilisateurs, ce qui donne une classe très difficile à utiliser. TBS est de plus en plus "strong" mais de moins en moins "tiny". Par ailleurs, la syntaxe de son gabarit souffre du même inconvénient que celle de Smarty.

VTemplate utilise des gabarits à la syntaxe redondante et comporte des bugs de fonctionnement. C'est dommage car il est sur la bonne voie.

ModeliXe comporte des bugs de fonctionnement.

Savant3 (non abordé ici) ([Lien42](#)) est très agressif envers Smarty, dont il est inspiré (je fais référence à la présentation du projet sur le site officiel), mais la syntaxe de son gabarit est encore plus complexe, donc encore moins utile. Certaines personnes le qualifient d'anti moteur de gabarits.

YATS (non abordé ici) ([Lien43](#)), héritier de FastTemplate, est très intéressant (puisque c'est une extension plutôt qu'une bibliothèque) mais le projet est inactif depuis trop longtemps, en fait depuis PHP 3.

7.4. Liens

À choisir parmi ces moteurs, je prends celui de phpBB2. Il permet une séparation correcte entre la logique applicative et l'affichage, il est facile d'accès et il fonctionne bien. De plus, il est maintenu et utilisé par une communauté gigantesque.

Si vous utilisez l'un ou l'autre moteur dans vos sites, n'hésitez pas à nous le faire savoir et à nous donner vos impressions (cf. lien ci-dessous vers le débat sur nos forums), surtout si elles sont différentes de celles proposées ici !

Ressources Développez :

Comparatif : Les systèmes de templates en PHP ([Lien44](#)), par Hugo Étievant ;

Tutoriel : Le système de templates de phpBB ([Lien45](#)), par Genova ;

Formation : Tous nos cours sur les gabarits en PHP ([Lien46](#));

Débat sur nos forums : quel système de gabarits utilisez-vous ? ([Lien47](#))

Retrouvez l'article de Guillaume Rossolini en ligne : [Lien48](#)

Récapitulatif des actualités Zend

Il y a du neuf chez Zend ces temps-ci... Après le Zend Studio 5.5 et la Zend Platform 3, les derniers jours ont vu la sortie de Zend Core 2.

Récemment, Julien annonçait sur son blog la disponibilité de la certification Zend (ZCE) en français. Vous pouvez désormais étudier en toute quiétude, la langue de Molière ne vous fera pas défaut dans cette étape importante de votre vie

Enfin, il est temps d'évoquer le Zend Framework.

La version bêta 0.9 est maintenant disponible, et Développez vous réserve une surprise à ce sujet dans les jours à venir. Cette version marque la fin des modifications structurelles et fonctionnelles jusqu'à la version 1.0 finale. Les prochaines versions seront des Release Candidates afin de corriger d'éventuels bugs, ce qui

signifie que le framework a atteint un stade de maturité jugé suffisant pour être utilisé dans vos projets professionnels. Attendez la version 1.0 finale pour mettre en production, mais vous pouvez désormais développer sans craindre de devoir modifier à nouveau votre code avant la version finale.

Pour rappel, l'équipe de Zend continue de nous aider à répondre à vos problèmes sur les forums, dans la mesure de leurs disponibilités. N'hésitez pas à poser toutes vos questions, c'est le moment !

--> Participer ([Lien49](#))

Retrouvez le billet en ligne : [Lien50](#)

Les derniers tutoriels et articles

Introduction à Windows Communication Foundation

A travers cet article nous allons découvrir le système des attributs de dotnet, voir à quoi ils servent, et comment créer ses propres attributs Cet article vous présente le principe de la communication entre applications et en particulier Windows Communication Foundation pour l'implémentation

1. Pourquoi faire des applications qui communiquent ?

C'est la base de tout : Pourquoi faire une application qui permet à une autre de communiquer avec elle ?

Assurément toute les applications n'ont pas besoin de communiquer avec les autres. Alors quand mettre en oeuvre une communication entre applications ?

- Si vous avez par exemple une application qui doit accéder à des ressources spécifiques à un poste.
- A des fins de centralisation de données
- A des fins de répartitions de charges sur plusieurs postes
- Afin de ne pas réécrire un traitement qui existe dans une autre application
- Pour pouvoir contrôler une application

Ce ne sont que des exemples mais ils résument bien les différents cas qui sont le plus souvent rencontrés

2. Windows Communication Foundation ? C'est quoi ?

Windows Communication Foundation (WCF) est la nouvelle couche de communication du framework 3.0. Cette couche a été créée afin d'unifier les différents modèles d'écritures d'applications "communicantes".

Unification des modèles de développement de communications
WCF fait partie du framework 3.0 qui est livré d'office avec Vista et est disponible sous Windows XP et 2003 serveur.

Pour pouvoir utiliser WCF dans vos applications vous devez ajouter la référence à System.ServiceModel à vos projets. Ensuite vous devez faire un using System.ServiceModel;

3. Comment fonctionne la communication entre applications ?

Afin que les applications puissent communiquer il faut qu'un certain nombres de contraintes soient respectées.

- Définir les méthodes exposées par le serveur
- Définir les types de données transmissibles entre applications
- Définir l'ABC de la communication

3.1. Définir les méthodes exposées par le serveur

Pour que le client puisse appeler des méthodes sur le serveur encore faut-il que ce premier connaisse les méthodes disponibles sur le serveur. WCF est très strict sur la façon d'implémenter la communication entre applications. En effet les méthodes exposées par WCF doivent toutes être définies dans une interface "décorée" par l'attribut ServiceContract. De plus les fonctions que l'on veut exposer par WCF doivent être "décorées" par l'attribut OperationContract.

```
using System.ServiceModel;
```

```
[ServiceContract]
```

```
public interface IMonPremierServiceWCF
{
    [OperationContract]
    void EcrireUnePhrase(string phrase);
}
```

Cette interface définit donc une méthode qu'un client peut appeler à distance. Au passage vous remarquerez que nous passons un paramètre sans aucun problème. Cette interface définit donc le contrat de notre service. En effet les types primaires tel que string, int, float, etc peuvent être utilisés sans aucun problème ni modification. Mais comment permettre à des types personnalisés d'être transmis par WCF d'un serveur à un client ?

3.2. Définir les types de données transmissibles entre applications

Afin de marquer une classe comme transmissible par WCF il suffit d'utiliser l'attribut [DataContract]. Cet attribut est défini dans l'assembly System.Runtime.Serialization qu'il faut donc ajouter à son projet. Ce choix d'implémentation est très intéressant car il permet de ne pas bousculer toute la modélisation d'un projet en obligeant l'héritage par une classe comme MarshalByRefObject en Remoting. Cela vous permet donc de migrer facilement et avec peu de modifications un projet qui n'as pas été initialement prévu pour fonctionner en WCF.

Une fois que votre classe est "décorée" avec l'attribut [DataContract] il faut ensuite identifier les méthodes que nous allons exposer aux futurs clients de notre application.

```
[DataContract]
public class MyDatas{
    private string data1

    [DataMember]
    public string Data1{
        get{
            return this.data1;
        }
        set{
            this.data1 = value;
        }
    }
}
```

3.3. Définir l'abc de la communication

Qu'est ce que l'abc de la communication ? C'est sous cet acronyme que Microsoft définit les 3 étapes essentielles du déploiement d'un service WCF.

- A pour Address : Définit l'adresse du serveur qui expose le service

- B pour Binding : Définit la façon dont le service sera exposé
- C pour Contract : Définit le contrat que le service remplit

La grande force de WCF est sans aucun doute la façon dont il a été pensé. En effet tout ce qui touche au déploiement et à la consommation d'un service WCF est défini dans le fichier de configuration de l'application. Cela permet de s'affranchir entièrement des contraintes de déploiement au moment du développement. En unifiant ainsi les différentes façons de programmer une application distribuée, WCF permet une souplesse et une facilité d'utilisation extrêmement agréables.

3.3.1. Définir l'adresse du service

Afin de pouvoir exposer un service vous devez définir l'adresse à partir de laquelle il sera accessible. Pour cela il faut définir les sections correspondantes dans le fichier de configuration de l'application. Ces sections doivent toutes se trouver dans <system.serviceModel> de votre fichier de configuration.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system.serviceModel>
    <services>
      <service
name="DvpDemoWCF.IMonPremierServiceWCF">
        <endpoint
address="net.tcp://localhost:5000/DvpDemoWCF"
binding="netTcpBinding"
contract="DvpDemoWCF.IMonPremierServiceWCF"/>
      </service>
    </services>
  </system.serviceModel>
</configuration>
```

l'attribut address vous permet de définir un point de terminaison c'est à dire une URI où le service sera accessible. De plus cela vous permet de configurer la totalité de l'abc de la communication.

Ici l'adresse expose le service sur le poste local sur le port 5000 et à l'adresse DvpDemoWCF

3.3.2. Définir le binding du service

Le binding est la façon dont le service sera exposé. C'est à dire qu'il vous permet de définir le protocole utilisé pour transporter vos objets sur le réseaux ou en local. Chaque binding à ses avantages et ses inconvénients mais au niveau du développeur ce choix ne doit pas être bloquant. En effet lors de votre développement vous ne savez pas toujours dans quel environnement votre application va être déployée. L'administrateur réseau ne veut peut être pas ouvrir un port sur un serveur pour des raisons de sécurité, ou alors il voudrait pouvoir définir quel port. De même il préfère peut être vu la topologie de son réseau que les communications se fassent en tcp afin de réduire la charge réseau. Toutes ces questions ne sont donc pas à la charge du développeur mais bien de l'administrateur réseaux qui va s'occuper de l'installation de votre application. C'est la que l'on comprend une autre des choses merveilleuses de WCF. En .NET Remoting ce choix était fait par l'équipe de développement et l'administrateur réseaux n'avait aucun moyen d'influer dessus. WCF de part l'uniformisation des techniques d'utilisation des protocoles permet de rendre ce choix à la personne concernée : l'administrateur réseaux. Par défaut le framework vous permet d'utiliser 4 méthodes d'exposition de service:

- basicHttpBinding, HTTP, XML 1.0, Interopérable
- wsHttpBinding, HTTP, XML 1.0, Sécurité,
- wsDualHttpBinding, HTTP, XML 1.0, Sécurité,

communication double sens

- netTcpBinding, TCP, encodage Binaire, Sécurité, communication double sens
- netNamedPipeBinding, tubes nommés, encodage Binaire, Local seulement, communication double sens
- netMsmqBinding, Message windows, encodage Binaire, Windows seulement

Le binding est donc défini par l'attribut "binding" de notre fichier de configuration.

3.3.3. Définir le contrat du service

Le "contract" (contrat en français) est tout simplement la liste des méthodes appellables par les clients. Rappelez vous, nous avons défini une interface que nous avons décoré avec les attributs [ServiceContract] et [OperationContract]. C'est cette interface que nous allons définir en tant que contrat pour notre client.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system.serviceModel>
    <services>
      <service
name="DvpDemoWCF.IMonPremierServiceWCF">
        <endpoint
address="net.tcp://localhost:5000/DvpDemoWCF"
binding="netTcpBinding"
contract="DvpDemoWCF.IMonPremierServiceWCF"/>
      </service>
    </services>
  </system.serviceModel>
</configuration>
```

A noter qu'un contrat est obligatoirement une interface. Vous ne pouvez pas définir une classe en tant que contrat car le framework vous impose une séparation forte entre ce que les clients peuvent voir et ce que le serveur implémente. Ceci est une bonne chose qui vous permettra à la longue de ne pas vous mélanger entre des classes qui sont appelable par le client, des classes qui ne le sont pas et des interfaces de définition de services.

3.4. Conclusion

A travers cette présentation générale de WCF, vous avez pu constater que le modèle de programmation est bien plus simple que celui de .NET Remoting.

Vous avez également découvert en quoi WCF peut vous être utile dans vos développements personnels.

Nous allons maintenant étudier 1 démonstration en profondeur afin d'appliquer les fondamentaux que nous venons de voir.

4. Etude de cas : Service de déploiement à distance

Voyons les spécifications de notre service d'installation de logiciel à distance.

Ce logiciel doit tourner en tant que service Windows, il doit être transparent pour l'utilisateur. Il doit pouvoir interagir avec l'utilisateur en cas de besoin. Il doit envoyer le rapport de l'installation au serveur afin de maintenir la liste à jour.

Cependant il ne s'agit ici que d'un cas d'étude, nous n'allons pas mettre en oeuvre le service windows mais écrire des applications consoles.

4.1. Structure du projet

Comme nous l'avons vu, WCF nous impose une certaine structure afin de fonctionner.

Notre projet se divise donc en trois projets au sens Visual Studio.

- SoftwareInstallerServerService : Programme console destiné à être installé sur un serveur afin de fournir aux clients les informations nécessaires à l'installation des logiciels.
- SoftwareInstallerClientService : Programme winform destiné aux clients. Il doit se connecter à intervalles réguliers au serveur afin de vérifier si des logiciels doivent être installés.
- SoftwareInstallerLibrary : Librairie contenant les classes communes et l'interface de définition du serveur.

Ces trois projets sont une base de travail mais l'on pourrait tout à fait imaginer que le service serveur soit implémenté en ASP.NET par exemple.

4.2. SoftwareInstallerLibrary

Nous allons commencer par implémenter la librairie de définition du serveur. Comme nous l'avons vu cette définition intervient sous forme d'une interface décorée par [ServiceContract].

```

/// <summary>
/// Interface de définition du service distant
/// </summary>
[ServiceContract]
public interface RemoteService{
    /// <summary>
    /// Identifie le client sur le serveur.
    /// </summary>
    /// <param name="clientInformation">Les
    informations d'identification du client</param>
    [OperationContract]
    void IdentifyClient(ClientInformation
    clientInformation);

    /// <summary>
    /// Obtient la liste des logiciels à installer
    sur le client
    /// </summary>
    /// <returns></returns>
    [OperationContract]
    List<SoftwareData> GetSoftwareList();

    /// <summary>
    /// Renvoie le rapport d'installation au serveur
    afin de centraliser l'information
    /// </summary>
    /// <param name="clientReport">le rapport du
    client sur l'installation des logiciels</param>
    [OperationContract]
    void UploadReport(ClientReport clientReport);
}

```

Nous avons donc défini notre interface. Nous avons aussi fait appel à deux classes personnelles que nous allons détailler. La première est SoftwareData :

```

/// <summary>
/// Classe représentant un logiciel devant être
    installé sur le poste client
    /// </summary>
    [DataContract]
    public class SoftwareData{

        private string name;
        private string installerAddress;
        private Version version;

        /// <summary>

```

```

/// Constructeur par défaut
/// </summary>
    public SoftwareData(){

    }

    #region Proprietes

    /// <summary>
    /// Obtient ou définit le nom du logiciel à
    installer
    /// </summary>
    [DataMember]
    public string Name{
        get{
            return this.name;
        }

        set {
            this.name = value;
        }
    }

    /// <summary>
    /// Obtient ou définit l'adresse de l'installeur
    /// </summary>
    [DataMember]
    public string InstallerAddress{
        get{
            return this.installerAddress;
        }

        set{
            this.installerAddress = value;
        }
    }

    /// <summary>
    /// Obtient ou définit la version du logiciel à
    installer
    /// </summary>
    [DataMember]
    public Version Version{
        get{
            return this.version;
        }

        set{
            this.version = value;
        }
    }

    #endregion
}

```

et la classe ClientReport :

```

/// <summary>
/// Classe définissant un rapport suite à
    l'installation de logiciel sur le client
    /// </summary>
    [DataContract]
    public class ClientReport{
        private DateTime date;

        [DataMember]
        public DateTime Date{
            get { return date; }
            set { date = value; }
        }
    }

```

```

private string clientName;

public string ClientName{
    get { return clientName; }
    set { clientName = value; }
}

private List<SoftwareInstallReport> installReports;

public List<SoftwareInstallReport> InstallReports{
    get { return installReports; }
    set { installReports = value; }
}
}

```

Vous remarquerez qu'il n'y a aucun code spécifique à WCF à part les attributs sur les classes et les membres que l'on souhaite exposer au client. Notre librairie partagée entre le client et le serveur est maintenant finie.

Nous passons donc à la création du serveur.

4.3. SoftwareInstallerServerService

Ce serveur est un peu particulier. En effet il tourne en tant que programme console sur le PC serveur où il est installé. Il va sans dire qu'un service windows serait plus adapté mais ce n'est pas le sujet dans cet article. Une bonne partie de l'initialisation du serveur WCF va donc se faire directement dans la méthode Main du programme, mais cela n'est pas une bonne architecture ;-). De plus notre programme n'est pas "security friendly". En effet il demande les droits administrateurs pour fonctionner.

4.3.1. Création de la base de WCF

Nous allons écrire le code nécessaire à l'initialisation et la configuration du serveur WCF.

Pour commencer voyons le code d'initialisation du service serveur.

```

public void Main(string[] args){
    remoteService = new RemoteService();

    //Publication de la classe en singleton
    host = new ServiceHost(remoteService);

    //Ouverture du canal de communication
    host.Open();
}

```

Et le code d'arrêt du serveur :

```

//Fermeture du canal de communication
host.Close();

```

Ces quelques lignes de code suffisent à créer un canal de communication et à se mettre en attente d'une connexion d'un client. Il faut maintenant écrire le code du service proprement dit :

4.3.2. L'implémentation du service WCF

```

[ServiceBehavior(InstanceContextMode =
InstanceContextMode.Single, ConcurrencyMode =
ConcurrencyMode.Single)]
public class RemoteService : IRemoteService{

    private Dictionary<string, ClientInformation>
clientInformations;
    private List<string> clientAuthenticated;

```

```

    /// <summary>
    /// Constructeur par défaut
    /// </summary>
    public RemoteService(){
        this.clientInformations = new
Dictionary<string, ClientInformation>();
        this.clientAuthenticated = new List<string>();

        //DEBUG
        ClientInformation ci = new ClientInformation();
        ci.ClientVersion = new Version("1.0.0.0");
        ci.Name = "dev01";

        this.clientInformations.Add(ci.Name, ci);
    }

    #region IRemoteService Members

    public void IdentifyClient(ClientInformation
clientInformation){
        if
(!clientInformations.ContainsKey(clientInformation.Nam
e)){
            throw new Exception("Client inconnu !");
        }

        if
(clientAuthenticated.Contains(clientInformation.Name))
            throw new Exception("Client déjà
authentifié");

        this.clientAuthenticated.Add(clientInformation.N
ame);
    }

    public List<SoftwareData>
GetSoftwareList(ClientInformation clientInformation)
    {
        if
(this.clientAuthenticated.Contains(clientInformation.Nam
e)){
            //DEBUG
            SoftwareData soft = new SoftwareData();
            soft.InstallerAddress =
"\\myserver\partage\setup.exe";
            soft.Name = "Test";
            soft.Version = new Version("0.0.0.1");

            List<SoftwareData> lists = new
List<SoftwareData>();
            lists.Add(soft);

            return lists;
        }
        else{
            throw new Exception("Client non
authentifié");
        }
    }

    public void UploadReport(ClientReport
clientReport){

    }

    #endregion
}

```

La seule chose importante dans cette classe c'est l'attribut [ServiceBehavior(InstanceContextMode = InstanceContextMode.Single, ConcurrencyMode = ConcurrencyMode.Single)]. Cet attribut permet de définir que

notre classe est prévue pour être publiée en tant que singleton et sur un seul thread. Le paramètre `ConcurrencyMode` permet de définir si un thread doit être créé à chaque fois qu'un client fait un appel ou si les appels doivent attendre que le précédent soit fini. Il existe une troisième option mais tout cela va plus loin que cette introduction.

4.4. SoftwareInstallerClientService

Il ne reste plus qu'à écrire le programme client de notre projet.

```
/// <summary>
/// Service client récupérant la liste des softs à
installer sur le client
/// </summary>
public partial class SoftwareInstallerClientService {
    private ChannelFactory<IRemoteService>
channelFactory;
    private IRemoteService remoteService;
    private System.Timers.Timer timer;
    private ClientInformation clientInformation;

    public event UpdateNeededProgramEventHandler
UpdateNeededProgram;

    public SoftwareInstallerClientService() {
    }

    public void Start() {
        channelFactory = new
ChannelFactory<IRemoteService>("SI");
        channelFactory.Open();

        remoteService = channelFactory.CreateChannel();

        clientInformation = new ClientInformation();
        clientInformation.ClientVersion = new
Version("1.0.0.0");
        clientInformation.Name = "dev01";
        clientInformation.UserName =
Environment.UserName;

        //Envoi de l'authentification au serveur
remoteService.IdentifyClient(clientInformation);

        //Timer toute les 5 minutes
timer = new System.Timers.Timer(6000);
timer.Elapsed += new
ElapsedEventHandler(timer_Elapsed);
}
```

```
        timer.Enabled = true;
    }

    void timer_Elapsed(object sender, ElapsedEventArgs
e) {
        List<SoftwareData> softs =
remoteService.GetSoftwareList(clientInformation);

        if (this.UpdateNeededProgram != null)
            this.UpdateNeededProgram(softs);
    }

    public void Stop() {
        timer.Enabled = false;

        channelFactory.Close();
    }
}
```

Le point important de notre client est la façon dont on va récupérer l'instance du service. En effet il existe plusieurs manières, la plus courante étant de passer par un générateur d'interface, outil également utilisé pour la consommation de web service. Mais je trouve cette façon de faire extrêmement non pratique. Je préfère largement déployer l'interface du service sur le client que régénérer des classes avec un outil (ici le SoapSuds fourni par le SDK du framework). Nous utilisons une classe du framework 3.0 qui s'appelle `ChannelFactory` qui vous permet de récupérer une instance correspondant à l'interface du contrat. Cette classe peut prendre un certain nombre de paramètres pour son constructeur mais le plus utile est de passer le nom du endpoint à utiliser pour créer cette instance. Ensuite un simple appel à `CreateChannel` vous récupérera tout ce qu'il faut pour utiliser de manière transparente votre service WCF.

5. Conclusion

Pour finir cet article d'introduction à Windows Communication Foundation je dirais que l'arrivée de cette nouvelle technologie est une très très bonne chose. Elle rationalise et permet de fédérer les différentes méthodes de communications inter applications qu'elles soient locales ou distantes. De plus la simplicité est au rendez-vous alors pourquoi s'en priver ? Par contre, il reste encore beaucoup de choses à connaître de ce fabuleux outil ;-).

Retrouvez l'article de Vicent Lainé : [Lien51](#)

C & C++



Les derniers tutoriels et articles

Bien débuter en Langage C

Vous désirez apprendre le Langage C ? Vous ne savez pas par quoi commencer ? Ce guide va vous aider dans vos premiers pas, du choix de l'éditeur jusqu'à celui du livre ou des tutoriels en ligne ainsi que de nombreux liens pour vous permettre de vous exercer et d'aller plus loin !

1. Qu'est-ce que le C ?

Créé au début des années 70 par Dennis Ritchie et Brian Kernighan, le Langage C reste encore aujourd'hui un des langages les plus utilisés au monde !

Ce langage de programmation est à la base des systèmes d'exploitations que nous connaissons aujourd'hui ou au moins du noyau de ces systèmes comme par exemple Unix/Linux. Le Langage C a justement été créé pour un seul et unique but au départ, développer un système d'exploitation (Unix) mais au fil du temps, grâce à sa puissance, il a été adopté par une large communauté de développeurs ce qui a permis au langage d'évoluer et surtout d'être standardisé.

Ce langage est multi plate-forme, c'est-à-dire qu'un programme que vous créez par exemple sous Linux pourrait tout à fait être re-compilé sous Windows, BeOS, Amiga, etc.... sans devoir changer grand chose (voir rien du tout) dans le code source si vous respectez les normes en vigueur ANSI/ISO.

2. Je débute :

Les débuts ne sont jamais très aisés, notamment en ce qui concerne le choix des livres, la recherche de tutoriels en ligne et dans le choix des éditeurs ou EDI (Environnement de Développement Intégré). Les catégories ci-dessous vous permettront de vous acquitter de cette tâche souvent très longue et désagréable pour les débutants, tout ce que vous avez à faire, c'est de suivre ces différentes parties !

2.1. Quels livres/tutoriels en ligne puis-je lire ?

- Liste des cours complets sur developpez.com ([Lien52](#))
- Initiation au Langage C - Emmanuel Delahaye ([Lien53](#))
- Codage en Langage C - Emmanuel Delahaye ([Lien54](#))

2.2. Quels livres puis-je lire ?

- Méthodologie de la programmation en C ([Lien55](#))

2.3. Quels sont les outils dont j'ai besoin ?

Il vous faut pour bien commencer, un EDI (Environnement de Développement Intégré). Plusieurs EDI sont à votre disposition mais, la plupart d'entre eux ne sont disponibles que pour un seul système d'exploitation.

C'est là que Code::Blocks ([Lien56](#)) se distingue des autres EDI

car il est portable, ce qui signifie qu'il est disponible sur au moins les systèmes les plus utilisés et plus connus soit Windows, Linux et Mac OS X.

Ci-dessous vous trouverez un lien direct de téléchargement de Code::Blocks pour chacun de ces systèmes:

Sous Windows, vous avez la possibilité d'utiliser la version dite "Stable" ce qui est souvent préférable: Téléchargement de la version stable: 1.0-RC2 ([Lien57](#))
Cette version est livrée avec la suite de compilation MingW !

Si votre système est un système Linux, vous n'aurez pas le choix, il faut prendre une version Nightly Build ce qui signifie que c'est une version de test, une version Beta si vous préférez mais elle sont très stable à l'heure actuelle.

On trouve ces versions dans le forum officiel de Code::Blocks dans la section Nightly builds.

Il vous suffit simplement d'entrer dans le post contenant la version la plus récente et de télécharger l'archive ou le paquet suivant votre distribution. Je vous conseille également de consulter régulièrement cette page du forum car c'est presque quotidiennement que sortent de nouvelles versions Nightly builds ([Lien58](#)) !

Voici un tutoriel pour l'installation de Code::Blocks sur Linux Ubuntu: Installation de Code::Blocks sous Ubuntu ([Lien59](#))
La procédure d'installation sur d'autres distributions devrait être relativement identique à celle-ci !

La version Mac OS X est disponible en deux versions, suivant la version de votre OS: Version pour Mac OS X 10.3 – Panther ([Lien60](#)) et Version pour Mac OS X 10.4 - Tiger ([Lien61](#))

Voici également une présentation et test de cet EDI : Test de l'EDI C/C++ Code::Blocks ([Lien62](#))

3. Je veux approfondir mes connaissances :

Vous avez acquis de bonnes bases en Langage C, vous désirez approfondir vos connaissances soit dans le C ANSI/ISO ou voir les normes POSIX ou encore vous plonger dans la programmation spécifique à Linux ? Les livres et tutoriels ci-dessous vous sont proposés par developpez.com !

3.1. Quels livres/tutoriels en ligne puis-je lire ?

- Liste des tutoriels hébergés sur developpez.com ([Lien63](#))
- Notes sur le Langage C - Emmanuel Delahaye ([Lien64](#))
- Concevoir un composant logiciel en C - Emmanuel Delahaye ([Lien65](#))
- Les types abstraits de données (ADT) - Emmanuel Delahaye ([Lien66](#))
- Notes sur les réseaux - Emmanuel Delahaye ([Lien67](#))
- Les Threads POSIX.1 - Emmanuel Delahaye ([Lien68](#))
- Comment se fabriquer des entrées solides en C - Emmanuel Delahaye ([Lien69](#))

3.2. Quels livres puis-je lire ?

- Le langage C - Norme ANSI, 2ème édition ([Lien70](#))
- Exercices corrigés sur le langage C - Solutions des exercices du manuel de langage C de Kerrighan et Ritchie ([Lien71](#))
- C en action ([Lien72](#))
- Maîtrise des algorithmes en C ([Lien73](#))
- Programmation système en C sous Linux : Signaux, processus, threads, IPC et sockets ([Lien74](#))

4. Où puis-je trouver des exercices à faire ?

Il est toujours dur de trouver des exercices relativement corrects, les liens qui vous sont proposés ci-dessous vont aussi bien vers des exercices en rapport avec les maths qu'au développement de petits jeux en mode console et divers problèmes à résoudre !

- Récréomath ([Lien75](#))
- TopDev ([Lien76](#))
- Franc - IOI ([Lien77](#))

Le protocole SMTP

1. Introduction

L'une des applications les plus utilisées d'Internet, est sans nul doute l'envoi de mails. Comme vous le savez sûrement, l'envoi de mail est régi par le protocole SMTP, qui signifie **Simple Mail Transfert Protocol**.

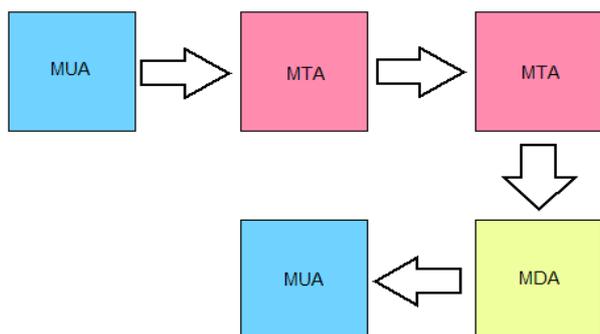
Mais savez vous exactement comment ce protocole fonctionne ? C'est ce que nous allons voir dans cet article.

Avec l'essor de ce protocole, une extension a été créée, l'**Extended SMTP (ESMTP)**. Il n'y a pas de grande différence entre les 2.

Il devient de plus en plus difficile de trouver un serveur ne supportant pas l'**ESMTP**. En conséquence tous les exemples suivants seront sur un serveur supportant **ESMTP**.

2. Cheminement d'un email

Voici le cheminement classique d'un email.



Vous pouvez également entreprendre un petit jeu (toujours en mode console) comme par exemple :

Mastermind : deux versions peuvent être créées :

La première serait une version simple où le joueur devine la combinaison mise en place par l'ordinateur.

La seconde serait une version où c'est cette fois-ci, c'est l'ordinateur qui devine votre combinaison.

C'est simplement qu'au lieu des couleurs on utilise alors des lettres ou des chiffres. Cet exercice est surtout intéressant du point de vue algorithmique et au niveau de l'IA !

5. J'ai tout lu mais il y a des choses que je n'ai pas comprises, que faire ?

Pas de panique ! Vous pouvez toujours poser vos questions sur les forums de developpez.com !

Voici les forums de developpez.com sur lesquels vous pouvez poser vos questions :

Forum : C ([Lien78](#))

Forum : Code::Blocks ([Lien79](#))

VII. Conclusion

J'espère que ce guide vous aura permis d'y voir un peu plus clair et de commencer votre apprentissage du Langage C plus sereinement.

L'équipe C/C++/Jeux vous souhaite une bonne programmation !

Si vous désirez aller encore plus loin dans la programmation en C et créer des interfaces graphique et des jeux, je vous invite à lire l'article complet de Frank H. en ligne : [Lien80](#)

Explication :

- **MUA** : Mail User Agent, c'est le client mail utilisé pour écrire le mail, le recevoir (Outlook, Thunderbird...).
- **MTA** : Mail Transfert Agent, c'est l'agent qui va transférer votre mail vers le MTA de votre destinataire.
- **MDA** : Mail Delivery Agent, c'est l'agent chargé de délivrer le mail à votre destinataire

Une fois votre mail parti depuis votre **MUA**, il est envoyé à votre **MTA**, qui est chargé de le délivrer au **MTA** de votre destinataire. Il se peut que ces derniers soit les mêmes, et il se peut aussi que le mail transite par plusieurs **MTA** avant d'arriver au dernier. Les serveurs **MTA** communiquent entre eux par le protocole **SMTP**, d'où leur nom de **serveur SMTP**.

Une fois sur le dernier **MTA**, ce dernier le transmet au **MDA**, qui stocke le mail avant que le destinataire ne vienne le relever.

À l'image du monde réel, le **MTA** représente le bureau de poste qui est chargé de recevoir tous les courriers, et d'ensuite les envoyer là où il faut, et si besoin est à un autre bureau de poste.

Le **MDA** représente la boîte aux lettres du client, où le courrier est stocké tant que le client n'est pas venu le relever.

Les serveurs **MDA** sont bien entendus protégés par des logins, mots de passes, pour éviter que tout le monde ne vienne lire le courrier de tout le monde. La récupération des messages se fait par les protocoles POP3/IMAP4 que nous verrons dans un prochain article.

3. Procédures SMTP

Voici les différentes procédures du protocole SMTP, de la simple ouverture de connexion, à l'envoi de mail. Nous allons voir les commandes associées à ces actions.

3.1. Code de retour

À chaque envoi de commande, le serveur nous renverra un code de retour, selon la réussite ou l'échec de la commande.

Voici les différents codes de retour et leur signification.

- 211 État système, ou réponse d'aide système
- 214 Message d'aide [Informations sur l'utilisation d'un récepteur ou signification d'une commande non standard particulière ; utile seulement pour un utilisateur humain]
- 220 <domaine> Service disponible
- 221 <domaine> Canal de transmission en cours de fermeture
- 250 Action de messagerie effectuée, succès
- 251 Utilisateur non local ; réémission vers <route-directe> (avec relais automatique)
- 354 Début du corps du message ; arrêt par <CRLF>.<CRLF>
- 421 <domaine> Service non disponible, canal en fermeture [Réponse à émettre sur tous les canaux lorsque le système exécute une séquence d'arrêt]
- 450 Action non effectuée : boîte aux lettres non disponible [Ex. : boîte aux lettres occupée]
- 451 Action arrêtée : erreur de traitement
- 452 Action non effectuée : manque de ressources système
- 500 Erreur de syntaxe, commande non reconnue [y compris des erreurs de type "ligne de commande trop longue"]
- 501 Erreur de syntaxe dans les paramètres ou arguments
- 502 Commande non implémentée
- 503 Mauvaise séquence de commandes
- 504 Paramètre de commande non implémenté
- 550 Action non effectuée : boîte-aux-lettres non disponible [Ex : boîte aux lettres non trouvée, pas d'accès]
- 551 Utilisateur non local ; essayer <route-directe> (sans relais automatique)
- 552 Action annulée : manque de ressources de stockage
- 553 Action non effectuée : nom de boîte-aux-lettres non autorisée [Ex : erreur de syntaxe dans le nom de boîte]
- 554 Transaction échouée

3.2. Ouverture et fermeture de liaison

Une fois la connexion au serveur effectuée, une vérification s'impose pour savoir si le client demandeur parle bien au serveur demandé.

Les 2 commandes suivantes sont donc utilisées à l'établissement et à la fermeture de la connexion.

```
HELO <domain> <CRLF>
```

Le domaine domain correspond à votre nom de domaine. Si vous n'en avez pas (ce qui est le cas de la plupart des internautes) vous pouvez mettre ce que vous voulez.

En réalité, le domaine importe peu, cette commande est juste là pour tester la relation.

Dans le protocole ESMTP (Extended SMTP), il faut envoyer EHLO

QUIT <CRLF>

Comme vous l'aurez deviné, QUIT quitte la connexion.

<CRLF> correspond à un "Carriage return" et à un "Life Feed" qui signifient respectivement "Retour Chariot" et "Nouvelle Ligne". En quelque sorte, cela correspond à un appui sur la touche Entrée de votre clavier.

Exemple d'ouverture de connexion :

```
220 mwinf2344.orange.fr ESMTP ABO
*****
EHLO skyrunner.home
250-mwinf2344.orange.fr
250-PIPELINING
250-SIZE 10485760
250-AUTH PLAIN LOGIN
250-AUTH=PLAIN LOGIN
250 8BITMIME
QUIT
221 Bye
```

Vous noterez qu'il s'agit d'un serveur ESMTP.

3.3. Emission de courrier

Une transaction SMTP se déroule en 3 étapes.

- La première donne l'identificateur de la transaction.
- La deuxième donne les destinataires.
- La troisième donne le contenu message.

La première étape se fait via cette commande

```
MAIL FROM:<adresse_source> <CRLF>
```

Si l'émetteur accepte la commande, il renvoie 250. adresse_source est l'adresse utilisée en cas de rapport d'erreur à transmettre (mail non délivré).

La deuxième étape est la liste des destinataires, via cette commande

```
RCPT TO:<dest> <CRLF>
```

Cette commande renvoie aussi 250, en cas de succès. Elle peut être répétée autant de fois que nécessaire, pour les différents destinataires, que ce soient des destinataires en copie ou copie cachée.

La troisième étape est donc la saisie et l'envoi du message via cette commande

```
DATA <CRLF>
```

Après cette commande, le récepteur SMTP renvoi 354.

Toutes les lignes saisies à la suite seront donc dans le message. Pour terminer la saisie du message, il faut taper une ligne avec un point unique.

Si vous voulez envoyer dans votre message une ligne, avec seulement un point, il faut en taper deux.

Après la saisie du message, le récepteur SMTP renvoi 250.

C'est aussi dans cette saisie que s'effectue l'envoi de l'en-tête du message (Date, Subject, From...), tout doit être spécifié manuellement. L'en-tête doit être spécifiée au début, et une ligne vide doit séparer l'en-tête, du corps du message. La structure d'un

message, appartient à elle seule à une RFC, la RFC822 . Si vous voulez plus d'infos sur cette dernière c'est par là RFC822 ([Lien81](#)) ou par là pour les anglophobes RFC822 ([Lien82](#))

Voici un exemple basique d'en-tête de message

```
Date: Fri, 29 Dec 06 14:36:00 +0100 GMT
From: "aaa" <aaa@aaa.com>
Subject: Test SMTP
To: "bbb" <bbb@bbb.com>
Cc: "xxx" <xxx@xxx.com>, "yyy" <yyy@yyy.com>
Bcc: "zzz" <zzz@zzz.com>
```

En-tête

- **Date** : date du message (possibilité d'antidater vos mails...). Si aucune date n'est spécifiée, le premier MTA s'en charge.
- **From** : indique l'expéditeur.
- **To** : indique les destinataires.
- **Cc** : indique les destinataires en copie.
- **Bcc** : indique les destinataires en copie cachée.
- **Subject** : sujet du message.
- Pour plus d'informations sur l'en-tête des messages, je vous renvoie à la RFC822.

4. Exemple via telnet

Le protocole SMTP à l'instar de HTTP, est un protocole basé sur le modèle client serveur, en mode texte.
Pour envoyer un mail, on peut donc facilement se connecter au serveur SMTP de son FAI, via telnet.
Le port de SMTP est le port 25.

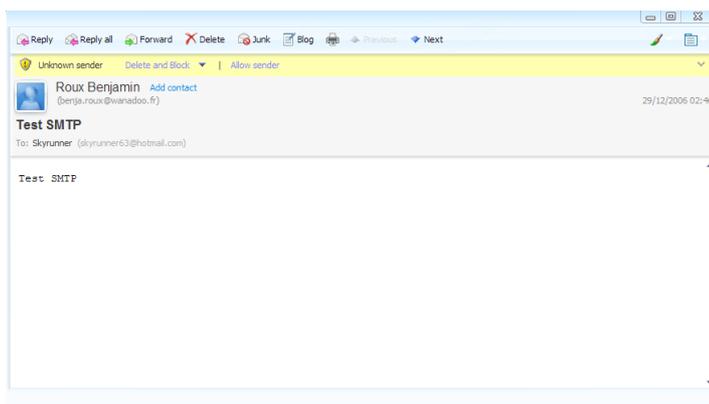
Si vous voulez refaire l'exemple ci-dessous, attention en utilisant telnet. En effet, lorsque vous tapez un caractère, il est directement envoyé, donc si vous vous trompez dans une commande, il faut la retaper.



Simple connexion à smtp.wanadoo.fr sur le port 25.



Envoi d'un mail, comme décrit dans la norme de SMTP, avec les bonnes commandes.



Le mail que j'ai reçu quelques secondes plus tard. Tout est bien rempli, y compris l'en tête (juste un petit problème d'heure à cause de l'heure d'hiver).

5. Conclusion

Vous venez donc de voir le protocole SMTP, qui est, comme vous avez pu le remarquer, un protocole extrêmement simple.

Vous pouvez donc dès maintenant faire vos propres programmes pour envoyer des e-mails.

Nous verrons prochainement le protocole POP, dans un prochain article.

Retrouvez l'article complet de Benjamin Roux en ligne : [Lien83](#)

Les livres C/C++

C++ en action

C++ est un langage riche et complexe, aux puissantes fonctionnalités : bâti autour du langage C, il est orienté objet et propose des fonctions en ligne, la surcharge d'opérateurs, etc. Il se retrouve aujourd'hui à la base de nombreux applicatifs et d'interfaces de développement graphiques. Cet ouvrage propose un ensemble de 171 solutions adaptables immédiatement sur le modèle : un problème, une solution, une discussion. Des experts en C++ répondent ici aux questions que tout programmeur en C++ se pose un jour ou l'autre. L'accent a été mis d'une part sur la

performance et sur la portabilité et d'autre part sur le recours aux nombreuses bibliothèques C++ et notamment Boost. Les recettes proposées couvrent un large éventail de domaines, allant des tâches simples à des manipulations plus complexes ; vous trouverez par exemple : Travailler avec des nombres pour les convertir, les comparer. Dates et heures : formatage, conversion, etc. Gestion des exceptions et sécurité : créer une classe exception. Classes et objets : initialiser des variables membres de classe, appeler une fonction virtuelle d'une super-classe. Construire des applications C++ avec make, Boost. Build. Analyser du XML et utiliser les Schemas, XSLT, XPath.

Manipuler des chaînes de caractères et du texte. Le multithreading : créer un thread, initialiser une seule fois une ressource partagée, etc. L'internationalisation : lire et écrire des devises, des dates, saisir une chaîne Unicode. Calculs mathématiques et scientifiques : somme, moyenne, statistiques, distance entre deux vecteurs.

Les algorithmes : retirer des objets d'un conteneur, ordonner un intervalle. S'il est parfois instructif de réinventer la roue, le faire en C++ n'apportera que désagréments et perte de temps. Recourir à la bibliothèque standard de C++, aux bibliothèques Boost et aux solutions proposées ici feront gagner un temps précieux au programmeur et lui épargneront des lignes de codes inutiles, qu'il soit débutant ou plus expérimenté.

Critique du livre par la rédaction (farscape)

Ce livre se veut pratique, il ne propose pas moins de 171 recettes utilisables et adaptables à nos besoins.

L'ensemble de ces solutions se veut standard en s'appuyant sur la STL ou sur d'autres bibliothèques comme Boost.

Chaque proposition est traitée selon le modèle : un problème, une solution et une discussion souvent assortie de conseils judicieux

Par exemple : pourquoi il est préférable dans une boucle de parcours d'utiliser la pré-incrémentation (++p) sur un itérateur au lieu de la post-incrémentation (p++).

Pourquoi privilégier l'accès aux données par un itérateur plutôt que par l'opérateur crochet dans le cas d'un vector.

Au fil des discussions et des présentations des différents conteneurs on prendra mieux conscience de l'impact de nos choix en matière de performance et surtout comment les améliorer.

Les exemples présentés sont simples et concis et bien commentés.

Le premier chapitre concerne la construction des programmes et de bibliothèques en C++ avec les principaux compilateurs en mode ligne de commande, l'installation de Boost et la création des bibliothèques dynamiques avec boost.build.

Passé ces étapes on entre dans le vif du sujet, chaque chapitre traite un sujet distinct et peut être lu séparément.

A travers tous les thèmes étudiés la bibliothèque standard STL est vraiment passée au crible avec, les conversions entre types, les manipulations sur les chaînes de caractères, les conteneurs, les itérateurs, les différents algorithmes, la gestion des exceptions, les flux et fichiers etc..

Au bout du compte ce livre est un excellent moyen pour s'initier, voir se perfectionner à la manipulation de la STL ou encore chercher une recette toute faite.

Enfin, j'ai trouvé très intéressant la mise en œuvre de Boost sur des sujets comme la conversion des chaînes en numérique, la

gestion du multithreading ou encore la sérialisation XML.

En conclusion voila un livre qui traite de sujets concrets à mettre en action dans notre quotidien de développement, et qui va trôner en bonne place sur mon bureau...

Retrouvez ce livre sur la rubrique C : [Lien84](#)

C++

Cet ouvrage traite de la programmation d'applications informatiques en langage C++ (à l'aide de l'environnement de développement Visual C++ 2005 Express Edition qui est gratuit). Tous les aspects du C++ sont traités, de la découverte du langage aux fonctionnalités avancées du C++ (templates, STL), à passant par les concepts de la programmation orientée objet... Mais le livre va également plus loin avec entre autre :

- la programmation .NET 2.0 en C++/CLI
- le développement d'applications fenêtrées Windows Forms
- le graphisme avec la bibliothèque GDI+
- l'initiation à la 3D temps réel avec OpenGL
- l'utilisation avancée du compilateur

l'ajout de composants et de bibliothèques de classes pour Visual... Le livre propose au lecteur de développer de nombreuses applications tout au long des chapitres, en partant de son premier programme console jusqu'au développement d'un logiciel de dessin multidocument, clone de Paint. Les auteurs délivrent également des conseils malins pour la programmation.

Critique du livre par la rédaction (Pierre Schwartz)

Ne vous y trompez pas, ce livre bon marché est un livre d'introduction à la programmation C++ sous Windows. Il vous présentera de manière rapide et globale toute une série d'outils permettant de vous lancer dans la programmation. Seul l'aspect Windows est traité, de même, seule la programmation sous Visual C++ Express est abordée, mettant de côté les différences de compilateurs, d'environnements et de systèmes.

Ce livre couvre un grand nombre de domaines allant de la simple programmation C++ au C++/CLI en passant par la STL, OpenGL, l'utilisation de Visual C++ Express... Impossible d'étudier en profondeur tous ces concepts en un seul livre, nous avons donc droit à des rapides survols agrémentés de nombreux exemples pour montrer les différentes possibilités des outils abordés. Ce livre vous permettra donc de vous faire une première idée sur tous ces domaines, libre à vous ensuite de poursuivre vos lectures avec des livres plus spécialisés.

Idéalement, ce livre est à conseiller aux étudiants qui souhaitent compléter leurs cours par des aspects plus applicatifs et éventuellement plus motivants, mettant de côté l'algorithmique pour se tourner vers les nombreuses possibilités offertes par le C++ sous Windows.

Retrouvez ce livre sur la rubrique C : [Lien85](#)

Les derniers tutoriels et articles

Déclaration P3P, pourquoi et comment.

L'utilisation du P3P: informer de votre politique de confidentialité
Le P3P (Platform for Privacy Preferences) est une initiative du W3C (World Wide Web Consortium) datant de 2002 et visant à normaliser les déclarations informatives relatives à la protection des données personnelles des internautes. En effet, trop de sites ne précisent pas ce qu'ils font de l'usage des données collectées ; or ces données ne résultent pas uniquement d'une collecte émanant d'un formulaire rempli sciemment par vos soins, mais sont souvent recueillis sans que vous en ayez connaissance ... En affichant clairement leurs politiques de confidentialité, les sites Web participent à installer une relation de confiance avec les internautes. Voici deux exemples qui devraient vous permettre de voir l'intérêt du P3P au niveau des internautes :

1. Cookies traceurs permettent à des sites tiers de suivre votre navigation sur le Web : Ils seront bloqués automatiquement par comparaison entre l'information envoyée par le serveur, via la déclaration P3P, et le paramétrage de votre navigateur.
2. Collecte des informations sur les mineurs à des fins intentionnelles : La mise en place d'un filtre parental n'est pas efficace si ce dernier se limite à des mots interdits. Il est préférable de se baser sur les informations retournées par le serveur, afin de limiter les risques inhérents à la collecte d'informations personnelles par les sites pédophiles notamment.

Actuellement, le P3P ne sert qu'à établir un réseau de confiance, les déclarations sont faites par les sites Web sous leur seule responsabilité. Afin d'éviter qu'un site malveillant ne fasse une fausse déclaration, une certification des politiques P3P est prévue, mais pas encore en vigueur. Néanmoins, cette pratique n'étant que peu généralisée, la quasi totalité des sites affichant de telles déclarations le fait dans un souci de transparence. Prendre le temps de déclarer son site est donc une bonne pratique. S'il existe deux formes de déclarations (compacte et complète) il est préférable d'utiliser la seconde qui, via un fichier XML situé dans un emplacement notoire, sera accessible aux navigateurs et aux internautes, contrairement à la forme compacte qui n'est qu'une suite de code accessible aux navigateurs et donc opaque pour l'utilisateur.

Pour procéder à la mise en place de la politique P3P, il convient de procéder en plusieurs étapes. Tout d'abord, afin de remplir de manière normalisée cette déclaration, le site de l'Organisation de coopération et de développement économiques (OECD) propose un formulaire à l'URL suivante : <http://www2.oecd.org/pwv3fr/> Ledit formulaire permet l'obtention d'un fichier `pwstatement.asp`. Ce fichier contient les informations qui vont vous servir pour construire votre déclaration XML. Vous devez donc ouvrir, via votre éditeur HTML, le fichier `pwstatement.asp`. Ce fichier est long et comporte environ 2800 lignes de code... beaucoup sont allouées au tableau de récapitulation. Il convient donc de prendre les informations pertinentes et qui permettent de répondre à des points utiles (exemples) :

- Autorisation d'accès anonyme - Vous pouvez accéder à

notre page d'accueil et consulter nos sites sans nous communiquer de données personnelles.

- Notre site web vous permet de communiquer avec d'autres visiteurs ou de publier des informations accessibles par d'autres internautes. Si vous utilisez ces fonctions, d'autres visiteurs peuvent collecter ces informations.
- Notre site web peut également comporter des liens avec Xiti, Urchin, Google AdSense, ces fournisseurs pourront collecter des données personnelles sur nos visiteurs.
- Collecte automatique d'informations - Nos sites web ne comprennent pas de cookies.
- Nous recourons à des moyens techniques (programmation par exemple) permettant de collecter automatiquement des données personnelles sur nos visiteurs ou bien nous sommes en mesure d'établir un lien entre des informations consignées automatiquement et des données nominatives concernant des personnes particulières. Nous procédons ainsi pour les raisons suivantes : Administration technique du site Web.
- Données collectées et spécification des finalités - Nous collectons des données nominatives si elles sont fournies volontairement lorsque vous utiliser nos services. Nous ne collectons pas d'informations provenant d'autres sources (fichiers ou organismes publics, organisations privées) sur nos visiteurs.
- Protection des données concernant les mineurs - Nous ne collectons pas délibérément de données personnelles sur des mineurs. Pour protéger la vie privée des mineurs sur lesquels nous collectons des données, nous prenons les mesures spécifiques suivantes : Nous nous efforçons, dans la mesure du possible, de vérifier qu'un parent a donné son consentement à la collecte de données personnelles sur le mineur. Sur notre page principale et à tous les points de notre site web où nous collectons délibérément des données sur les mineurs, nous présentons des informations sur nos pratiques en matière de données personnelles relatives aux mineurs.
- Nous ne divulguons pas vos données personnelles à nos filiales ou à d'autres organisations.
- Nous ne donnons pas la possibilité aux visiteurs de notre site d'utiliser une méthode de transmission sécurisée pour nous envoyer des données personnelles.
- Notre site obéit à une politique, à des règles ou à des mesures de sécurité visant à protéger les renseignements que nous détenons sur les visiteurs contre les risques suivants : Accès non autorisé ; Utilisation ou divulgation abusives ; Modifications ou altérations non autorisées ; Destruction illicite ou perte accidentelle.
- Nous garantissons que vos données personnelles ne seront pas communiquées à des institutions ou à des autorités gouvernementales, hormis dans les cas prévus par la loi ou la réglementation.
- Notre politique de la vie privée est conforme aux normes

suivantes : Lignes directrices de l'OCDE pour la protection de la vie privée et les flux transfrontières de données de caractère personnel ; Principes directeurs des Nations Unies pour la réglementation des fichiers personnels informatisés.

- Une fois ces informations mises de côté dans un fichier.txt, comme ci-dessus, il convient tout d'abord de créer un répertoire w3c à la racine de son site. Si d'autres emplacements sont possibles, il est vivement conseillé de respecter cet emplacement notoire, tel que définit par le W3C - cf traduction française de spécification de la plateforme pour les préférences de confidentialité.

Il faut ensuite créer le fichier p3p.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<?xml-stylesheet href="/p3p.xsl" type="text/xsl"?>
<META xmlns="http://www.w3.org/2000/12/P3Pv1">
  <POLICY-REFERENCES>
    <EXPIRY max-age="604800"/>
    <POLICY-REF about="#sitepolicy">
      <INCLUDE>/**</INCLUDE>
    </POLICY-REF>
  </POLICY-REFERENCES>
  <POLICIES>
    <POLICY name="sitepolicy" discuri="url
de la déclaration accessible au public">
      <ENTITY>
        <DATA-GROUP>
          <DATA
ref="#business.name">Nom de domaine</DATA>
          <DATA
ref="#business.contact-info.online.email">adresse e-
mail</DATA>
          <DATA
ref="#business.contact-info.postal.country">Pays</DATA>
        </DATA-GROUP>
      </ENTITY>
      <ACCESS>
        <ident-contact/>
      </ACCESS>
    </POLICY>
  </POLICIES>
</META>
```

Nota : Il importe de respecter la casse. Il est possible d'indiquer d'autres DATA que les 3 indiqués ci-dessus :

```
<DATA ref="#business.contact-
info.postal.street">Adresse postale</DATA>
<DATA ref="#business.contact-
info.postal.city">Ville</DATA>
<DATA ref="#business.contact-
info.postal.stateprov">Département</DATA>
<DATA ref="#business.contact-
info.postal.postalcode">Code postal</DATA>
<DATA ref="#business.contact-
info.telecom.telephone">téléphone, en international :
+33 (0) 000 000 000</DATA>
```

La ligne numéro 2 du fichier p3p.xml indique que le fichier p3p.xsl vient s'ajouter au contenu structurel du fichier xml ; p3p.xsl contient le contenu de votre déclaration de confidentialité :

```
<?xml version="1.0" ?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:p3p="http://www.w3.org/2000/12/P3Pv1"
xmlns="http://www.w3.org/1999/xhtml" exclude-
result-prefixes="p3p">
```

```
<xsl:template match="/">
  <xsl:apply-templates select="p3p:META"/>
</xsl:template>
<xsl:template match="p3p:META">
  <html>
    <head>
      <title>Politique de protection de la vie
privée</title>
      <link rel="stylesheet"
href="p3p.css" type="text/css" />
      <link rel="P3Pv1" href="url de la
déclaration accessible au public" />
    </head>
    <body>
      <h1>Politique de protection de la vie
privée de Nom de domaine</h1>
      <xsl:apply-templates
select="p3p:POLICIES"/>
      <xsl:apply-templates
select="p3p:POLICIES/p3p:POLICY"/>
      <hr />
      <address>
        <a href="url de la page
d'index de votre site">Nom de domaine</a>
        <a href="url de la page
de contact de votre site">Contact</a>
      </address>
    </body>
  </html>
</xsl:template>
<xsl:template match="p3p:POLICIES">
  <p>Ce fichier décrit la politique de protection
de la vie privée de Nom de domaine</p>
</xsl:template>
<xsl:template match="p3p:POLICY">
  <h2 class="policy" id="{./@name}">Eléments de la
politique de protection de la vie privée</h2>
  <xsl:variable
name="refpath">#<xsl:value-of
select="./@name"/></xsl:variable>
  <xsl:apply-templates
select="//p3p:POLICY-REF[@about=$refpath]"/>
  <dl>
    <xsl:apply-templates
select="p3p:ENTITY"/>
  </dl>
</xsl:template>
<xsl:template match="p3p:POLICY-REF">
  <!--Les déclarations suivantes ne le sont qu'à
titre d'exemple, pour en savoir plus,
il convient de préremplir la déclaration de
l'OECD et de regarder le code source du
fichier asp généré.-->
  <p>Vous pouvez accéder à notre page d'accueil et
consulter notre site sans nous
communiquer de données
personnelles.</p>
  <p>Le formulaire d'inscription à la newsletter
du site vous demande des données
personnelles nécessaires pour pouvoir vous
contacter et vous offrir le meilleur service
possible. Nous nous engageons à protéger ces
données et à ne pas les communiquer à une
tierce personne, sauf si la loi nous
l'autorisait.</p>
  <p>Conformément à la loi "Informatique et
Libertés" du 6 janvier 1978, vous disposez
d'un droit d'accès, de modification, de
rectification et de suppression des données vous
concernant. Vous pouvez pour cela nous contacter
via la page contact.</p>
  <p>Nom de domaine est enregistré à la <acronym
title="Commission Nationale de
l'Informatique et des Libertés">
CNIL</acronym> sous le numéro 0000000.</p>
</xsl:template>
```

```

<xsl:template match="p3p:ENTITY">
  <xsl:apply-templates select="p3p:DATA-
GROUP"/>
</xsl:template>
<xsl:template match="p3p:DATA-GROUP">
  <dl>
    <xsl:for-each select="p3p:DATA">
      <dd>
        <xsl:choose>
          <xsl:when test="./p3p:CATEGORIES">
            <xsl:for-each
select="p3p:CATEGORIES/*"><xsl:value-of select="local-
name()"/></xsl:for-each>
          </xsl:when>
          <xsl:otherwise>
            <xsl:value-of select="."/>
          </xsl:otherwise>
        </xsl:choose>
      </dd>
    </xsl:for-each>
  </dl>
</xsl:template>
</xsl:stylesheet>

```

```

body{
  color:#fff;
  background-color:#000;
  padding:10px;
  margin:2em auto 0 auto;
  width:760px;
  font:0.98em Arial, helvetica, sans-serif;
}
a{
  color:#fff;
  background-color:#000;
}
a :hover{
  text-decoration:none;
}
a :visited{
  text-decoration:line-through;
}

```

Le contenu de la déclaration étant réalisée, il convient de la mettre sommairement en forme, et de créer une feuille de style p3p.css (appelée en ligne 11 de p3p.xml) :

Votre déclaration P3P est maintenant terminée et il ne vous reste plus qu'à la soumettre au validateur du W3C ([Lien86](#)) Votre site n'en sera pas meilleur pour autant qu'un autre, mais vous participerez à la généralisation d'un Web de qualité, et un site de qualité est le meilleur atout pour intéresser les internautes ;o)

Les livres XML

XUL

A travers la création d'un forum de discussion, découvrez et maîtrisez la puissance de l'environnement de développement Mozilla, depuis la réalisation d'interfaces graphiques avec XUL et CSS jusqu'à la logique applicative avec Javascript et XPCOM et le déploiement avec XPI. On verra comment intégrer l'application dans le navigateur grâce aux overlays et à RDF et comment recourir à l'API DOM dans Javascript et PHP 5. L'étude de cas décrit également l'interfaçage avec un annuaire LDAP, la connexion à un serveur SOAP ou la réutilisation de code avec XBL.

Critique du livre par la rédaction (Erwan Amoureux)

Une bonne présentation du domaine XUL ,au langage comme aux autres technologies attenantes pouvant être utilisé. Ce livre permet de faire connaissance avec l'ensemble du panel et des différentes possibilités. Avec ces nombreux renvois et encarts , il permet de se faire un très bonne idée du domaine.

Le choix de l'exemple est peut être moins probant. Si l'exemple , un forum, à le mérite d'être connu de tous (et d'être un exemple "classique"), l'utilisation d'un XUL/PHP n'y semble pas particulièrement plus intéressante que le classique HTML/PHP, les principaux avantages de XUL (client lourd ou semi lourd), n'y étant guère valoriser. Cette partie reste néanmoins très bien construite pour apprendre ce langage.

Peut-être une petite faiblesse dans la dernière partie ou , si l'inventaire des technos attenantes est très bien faites, celui-ci aurait gagné en clarté à être un plus étoffé en exemple et explications.

En résumé, un excellent livre pour débutant, ou toutes personnes désirant prendre connaissance avec cette technologie, mais qui devra être complété par d'autres documentations pour tout approfondissement.

Retrouvez ce livre sur la rubrique XML : [Lien88](#)

XML en concentré

XML en concentré répond aux besoins des webmasters comme des développeurs en traitant les règles fondamentales de tout document XML. Cet ouvrage, complet et aussi concis que possible, détaille les règles de syntaxe les plus élémentaires ainsi que la création de DTD, les API à utiliser pour lire et écrire des documents XML dans divers langages de programmation.

Critique du livre par la rédaction (Nouridine Falola)

Cet ouvrage est composé de quatre parties. Les trois premières concernent l'apprentissage du XML et des technologies gravitant autour du XML. Assez bien adaptées aux débutants (d'aucun pourra trouver à juste titre ce livre un peu ardu pour commencer - dans ce cas on pourra trouver son bonheur avec "Introduction à XML"), ces parties ne seront pas inutiles en tant que révisions et approfondissement pour les connaisseurs. Les chapitres sont bien fournis en exemples souvent concis, et rigoureux tant la syntaxe doit être respectée (et donc comprise!).

La quatrième partie, d'environ 300 pages, constitue le manuel de référence. Ce manuel couvre XML 1.0 et 1.1 ainsi que la plupart des technologies XML les plus abouties (et utilisées) à ce jour. Ce manuel est indispensable à tout programmeur souhaitant retrouver une syntaxe.

En bref, beaucoup de rigueur et un bon niveau technique pour cet ouvrage. Un livre bien pensé, pour les débutants qui s'accroche comme pour ceux ayant déjà des connaissances en la matière. On pourra regretter cependant l'absence de chapitres consacrés à SVG ou MathML.

Retrouvez ce livre sur la rubrique XML : [Lien89](#)

Les derniers tutoriels et articles

Découvrez les nouveautés de Microsoft Access 2007

Découvrez la nouvelle version de Microsoft Access 2007.

A travers cet article, je vous propose de connaître l'avis des utilisateurs quant à la plupart des nouvelles fonctionnalités.

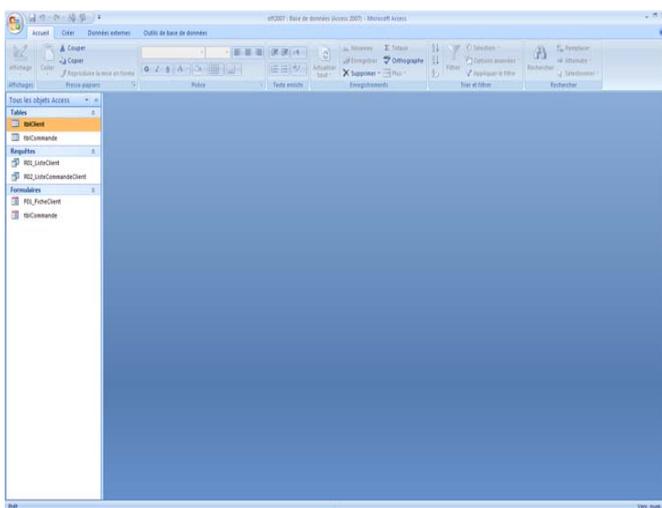
1. Introduction

Il y a dix ans sortait Microsoft Office 97, amenant de ce fait la plus populaire des versions de Microsoft Access. Aujourd'hui, en 2007, l'équipe Office renouvelle l'exploit et nous dévoile Microsoft Access 2007.

Après un an d'utilisation sur les versions bêta, je vous propose de vous faire découvrir une majeure partie des nouveautés du célèbre système de gestion de base de données.

2. La nouvelle interface

Si vous ne l'avez pas encore testé, voici à quoi ressemble Microsoft Access 2007 :

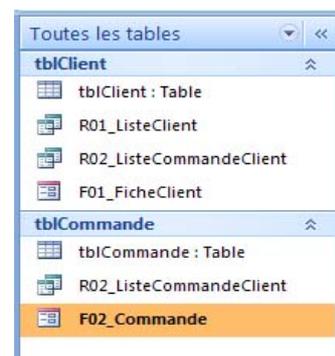


Bien entendu, le changement radical de l'interface graphique a des répercussions tant sur le développeur que sur l'utilisateur.

2.1. Le développeur

La nouvelle interface de la suite Office est axée sur l'utilisateur. Le développeur étant le premier utilisateur d'Access, Microsoft a amélioré l'ergonomie de son produit afin de rendre les développements plus rapides, plus précis, plus efficaces.

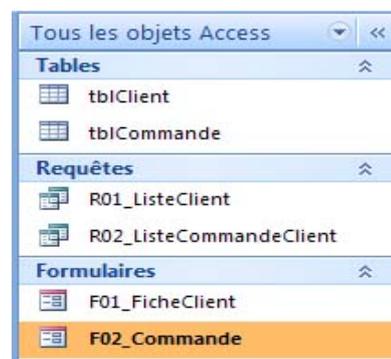
En outre, le nouveau **panneau de navigation** permet en un seul coup d'œil de déterminer les relations entre les différents objets de la base de données.



Dans l'exemple ci-dessus, le panneau regroupe les objets par dépendance. La requête R02 affichant les commandes par client, elle apparaît logiquement dans la hiérarchie des objets tblClient et tblCommande.

D'autres modes de tris sont disponibles : par date de création, par date de mise à jour, par type, etc.

Il est ainsi possible de retrouver le classement des anciennes versions :



D'autre part, ce panneau peut être replié en cliquant sur la flèche du coin supérieur gauche, agrandissant ainsi l'espace de travail. Grâce à cet outil, le développeur peut enfin suivre la totalité de son projet sans risque de se voir multiplier les objets orphelins ou inutiles.

Comme dans toute autre application Office, le ruban devient l'élément central de l'application. Toutes les fonctionnalités y sont regroupées. Là où avant une succession de plusieurs clics étaient nécessaires (au risque de se perdre dans les différentes pages d'options), il ne faut désormais plus que deux à trois actions pour activer, mettre en forme, créer, dessiner, exporter...

Les différents onglets contextuels permettent de restreindre l'affichage des commandes applicables à l'élément actif à l'écran.

A cause de la complexité des menus, Microsoft estime à 10 – 20 % le nombre de fonctionnalités réellement utilisées dans les versions précédentes de sa suite Office. Qui sait, le développeur Access va peut être découvrir des options qui existent déjà depuis 10 ans.

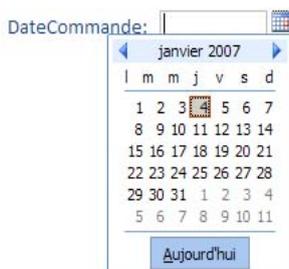
2.2. L'utilisateur

Si d'ores et déjà, le développeur est gâté par la nouvelle interface, l'utilisateur n'en est pas en reste. Un grand nombre de nouveautés viennent améliorer l'ergonomie des produits qu'il va utiliser (des produits que le développeur va construire). Sans oublier l'efficacité de l'incontournable ruban.

2.2.1. Les nouveautés

2.2.1.1. Le contrôle calendrier

A en juger le nombre de discussions sur le forum, les problèmes de sélection de dates sont monnaies courantes. Comment garantir efficacement que l'utilisateur saisit une date valide dans le format spécifié localement sur la machine ? Jusqu'à présent, le seul moyen était d'utiliser le contrôle Active X Microsoft Calendar. Bien entendu, comme pour tout ActiveX, la portabilité en termes de déploiement n'était pas au rendez-vous. Le problème est résolu avec un assistant à la saisie (intégré nativement dans les zones de texte). La nouvelle propriété Afficher Sélecteur de dates des zones de texte permet de définir si le calendrier doit être affiché ou non.



2.2.1.2. Optimisation du mode feuille de données

Le mode feuille de données permet de fournir une vision globale des données à l'utilisateur. Désormais, il est possible d'ajouter une ligne de synthèse en bas du tableau. L'utilisateur pourra ainsi consulter la somme, la moyenne, etc. des enregistrements présents dans sa table. Il pourra d'ailleurs changer lui-même la formule de calcul suivant ses besoins.

NumComan	NumClient	DateComm	Montant
1	1	11/01/2007	200,00 €
2	1	12/01/2007	30,00 €
3	2	09/01/2007	15,00 €
*	(Nouv.)		
Total			245,00 €

D'autre part, les nouveaux formulaires prennent en charge le double affichage permettant ainsi d'afficher la grille de données dans un formulaire classique.

Num :

Nom :

Prenom :

Adresse :

NumComan	DateComm	Montant
1	11/01/2007	200,00 €
2	12/01/2007	30,00 €
*	(Nouv.)	

Enr : 1 sur 2 | Aucun filtre | Rechercher

Le développeur peut choisir la place de la grille de donnée, déterminer si les préférences de l'utilisateur doivent être conservées, etc. Evidemment, les déplacements dans la grille sont synchronisés avec ceux du formulaire. Pour couronner le tout, il est même possible d'interdire la modification des données depuis le tableau.

2.2.1.3. Edition de texte enrichi

Qui n'a jamais rêvé de personnaliser la saisie dans les zones de textes ? Mettre quelques mots en gras, d'autres en rouge, etc. Jusqu'à présent il fallait utiliser le contrôle ActiveX RichTextBox. Désormais, les zones de texte prennent en charge cette fonctionnalité. Il suffit pour cela de définir la propriété Format du Texte à Texte Enrichi. La zone de saisie adopte alors le comportement de Microsoft Word.

Nom : Calibri 11

Prénom : G I S A ab

Adresse :

La mise en forme des données est stockée dans un champ mémo de la base de données sous forme de code HTML.

```
<div>Ceci est un <font color=red>essai</font></div>
```

Enfin, VBA offre une méthode permettant d'épurer le contenu du champ et de retourner l'information sans mise en forme.

```
Dim orst As DAO.Recordset
Set orst = CurrentDb.OpenRecordset _
("SELECT Texte from TblClient WHERE numclient=4")
Debug.Print Application.PlainText(orst.Fields(0))
```

Affiche :

Ceci est un essai

2.2.1.4. Les filtres de formulaires

Les filtres de formulaires ne sont pas une nouveauté mais ont subi une réelle amélioration. En cliquant avec le bouton droit de la souris sur un champ de formulaire, Access propose un menu contextuel permettant d'accéder aux enregistrements semblables, différents, supérieurs, inférieurs, etc.



La barre de navigation des formulaires accueille désormais une zone de saisie permettant d'effectuer une recherche dans tous les champs du formulaire et de se positionner sur les enregistrements correspondants.

2.2.1.5. Images dans les formulaires continus

L'insertion d'image dans les formulaires continus était un réel besoin des utilisateurs poussant les développeurs à recourir à des acrobaties vertigineuses pour combler ce déficit. C'est notamment le cas de Cafeine (MVP Access) dans son tutoriel : Repousser les limites des formulaires continus. De par sa complexité, cette démarche ne pouvait pas s'inscrire dans l'optique d'un produit « tout public ». Microsoft a donc résolu le problème en portant la gestion des images des formulaires uniques aux formulaires continus. Même si cette fonctionnalité était un peu fragile il y a encore quelques mois dans la bêta 2, elle semble aujourd'hui complètement opérationnelle.

2.2.1.6. Les variables temporaires

La nouvelle collection Application.TempVars permet de stocker des variables utilisables partout dans l'application et qui seront détruites à la sortie du programme. Chaque variable est identifiée par un nom unique dans la collection. L'ajout d'une variable déjà existante provoquera l'écrasement de la plus ancienne.

Voici les méthodes disponibles :

```
'Crée la variable MaVariable
TempVars.Add "MaVariable", 15
'Affiche 15
MsgBox TempVars("MaVariable")

TempVars.Add "MaVariable", 16
'Affiche 16
MsgBox TempVars("MaVariable")

'Supprime la variable MaVariable
TempVars.Remove "MaVariable"

'Supprime toutes les variables temporaires
TempVars.RemoveAll
```

Cette syntaxe évite le recours à des variables publiques et le nommage au sein de la collection rend leur accès très souple.

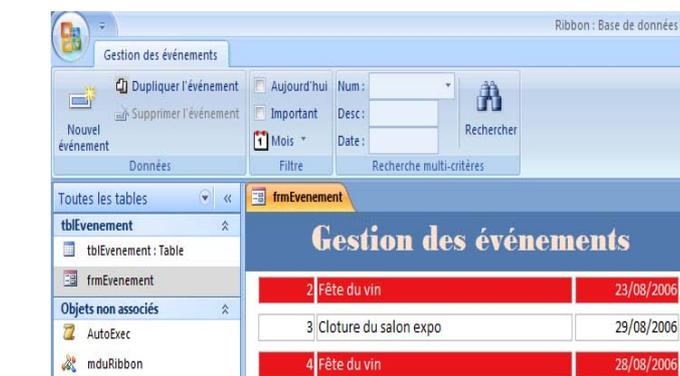
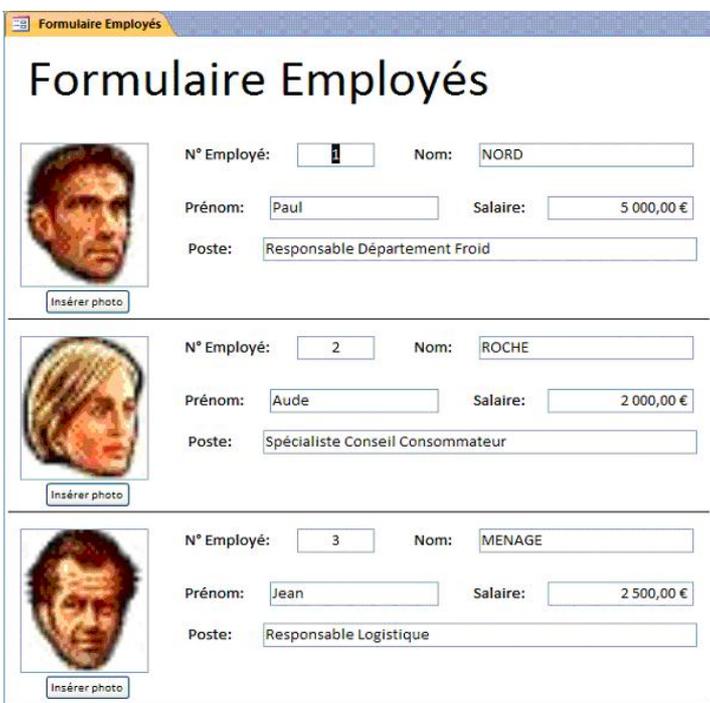
2.3. La programmation du ruban

Mon coup de cœur pour cette nouvelle version.

En mode exécution le ruban semble complètement inutile. Les fonctionnalités qu'il propose sont superflues. Pour ces raisons, le développeur aura souvent comme intention de le faire disparaître. Il s'agit là d'une très mauvaise idée. L'utilisateur est habitué aux produits Microsoft Office et a, de ce fait, pris un certain nombre d'habitudes. En toute logique, il s'attend à retrouver dans le ruban les commandes de l'application qu'il utilise.

Grâce à un fichier XML et quelques méthodes VBA vous allez pouvoir le transformer en un véritable centre de pilotage de l'application que vous avez développée.

Vous pouvez par exemple mettre en place un véritable assistant de recherche multi-critères :



Le formulaire retrouvera alors sa fonctionnalité première : l'affichage des données. Coté ergonomie, l'utilisateur pourra en un seul coup d'œil déterminer les parties métier et données de l'application. D'après les premiers retours que j'ai eus, il se sent guidé et aiguillé.

Pour comprendre le fonctionnement du ruban, vous pouvez consulter ces deux tutoriels :

- Programmation et personnalisation du ruban sous Access

- Gestion de favoris sous Access 2007 ([Lien92](#))

3. Travail collaboratif

3.1. Sharepoint Service

Microsoft SharePoint est le nouveau fer de lance de Microsoft pour les entreprises. Portail destiné à gérer l'activité de la société, il s'intègre naturellement dans la suite bureautique Microsoft Office.

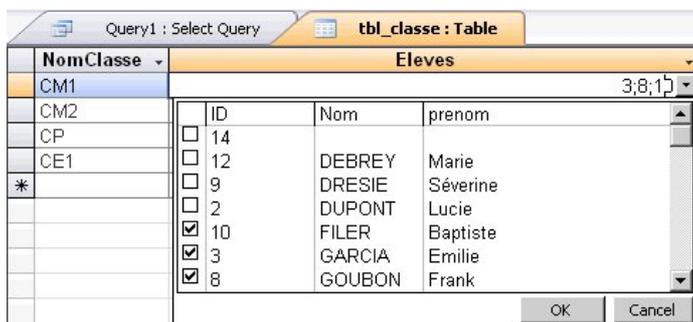
Dans SharePoint, les données sont stockées dans des listes (composants très proches des tables). Ces listes peuvent être importées ou liées dans Access 2007 et les tables de la base de données locale sont synchronisables avec le portail via les commandes de l'onglet Données Externes.

Si vous vous intéressez d'un peu plus près à SharePoint, vous vous apercevrez qu'il prend en charge certaines fonctionnalités inexistantes dans les versions antérieures d'Access. En voulant faire de SharePoint le cœur du système d'informations, Microsoft a été contraint de porter ces fonctionnalités aux produits Office.

Parmi ces nouveautés, deux ont retenues mon attention : les champs à valeurs multiples et les pièces jointes.

3.1.1. Les champs à valeurs multiples

Le champ multi-valué sème le trouble parmi les experts de la modélisation. Un tel champ est susceptible de recevoir plusieurs valeurs pour la même ligne (remplaçant ainsi une table de jointure), le tout avec une interface d'ajout des données plutôt sympathique :



Mais attention, son utilisation ne doit pas être un moyen de cacher une erreur conceptuelle ni même de simplifier les règles de gestion. Car en effet, s'il est adapté à un petit volume de données, il ne doit pas être utilisé dans une structure complexe et volumineuse sous peine de dégrader considérablement les performances.

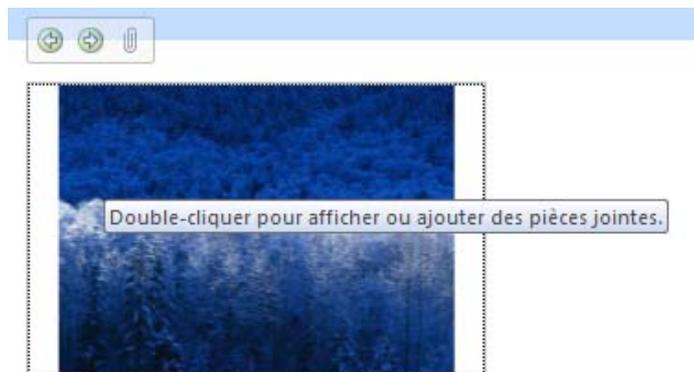
3.1.2. Les pièces-jointes

Les nouveaux champs de type pièces-jointes permettent de stocker des fichiers dans la base de données. Cette technologie repose sur les champs à valeurs multiples : il est possible d'ajouter plusieurs fichiers dans un champ pour un même enregistrement.

Contrairement aux champs OLE, le champ pièce-jointe est facilement utilisable en VBA : DAO intègre des méthodes permettant d'ajouter et d'enregistrer les fichiers.

Le nouveau contrôle pièce-jointe offre une interface ergonomique pour gérer ces fichiers dans les formulaires et permet une prévisualisation directe des images (idéal pour une galerie de

photos par exemple) . De plus, il est entièrement piloté en VBA.



D'autres informations sur les champs pièces-jointes sont disponibles dans le tutoriel : A la découverte d'Access 2007 : Les pièces jointes par l'exemple ([Lien93](#)).

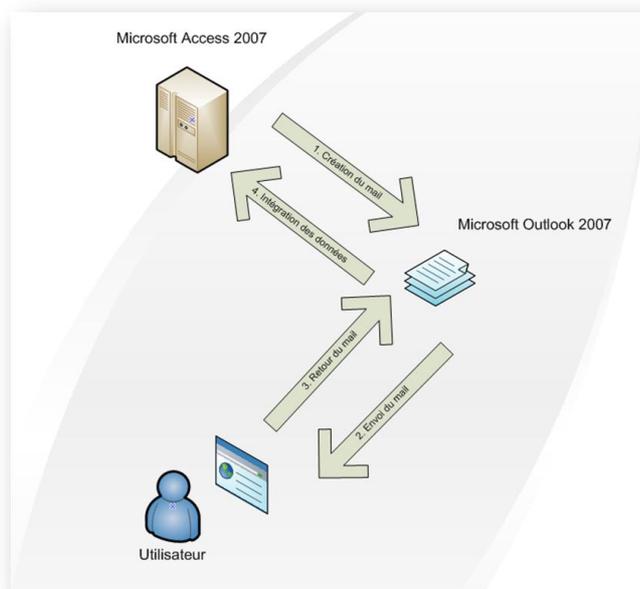
3.2. Outlook

Comme cela a été évoqué dans les sessions Web TV, Microsoft Outlook devient l'élément fédérateur des différentes applications de la suite Office. Ainsi, il intègre nativement les tâches MS Project, et tout autre élément Office. S'il est possible d'importer les données des dossiers Outlook dans Access, il est aussi possible de le transformer en collecteur de données pour vos tables.

Le processus est simple :

Access crée un formulaire avec des champs à remplir dans un courrier électronique. Outlook envoie ce mail aux utilisateurs. Via leur gestionnaire de messagerie, ils remplissent les différents champs et le retourne à l'adresse d'expédition. Outlook 2007 reçoit le mail et intègre automatiquement les données dans la table.

La seule limite réside dans la compatibilité des gestionnaires de messageries client : ils doivent impérativement être compatibles HTML sous peine d'être dans l'incapacité d'afficher le courrier.



A noter aussi que ce mode de collecte est également compatible avec les formulaires Microsoft InfoPath.

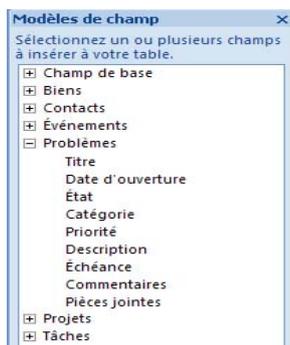
4. Portabilité et interopérabilité

4.1. Le format XML

L'interopérabilité des applications Office passe par la prise en charge intensive du format XML : OpenXML. Bien que l'export XML soit pris en charge dans les versions antérieures de Microsoft Access, il est désormais possible d'exporter et importer la structure des tables dans des fichiers XSD. Vous pouvez ainsi échanger les fichiers décrivant votre base de données à d'autres développeurs sans fournir les données, et vice-versa, favorisant de ce fait les échanges de contenu entre plusieurs applications.



Mieux, encore : définissez vos propres modèles de champs et de tables réutilisables par votre équipe. Vos développeurs créeront ainsi des champs possédant les mêmes règles de gestion. Pour cela rien, de plus simple. Il suffit d'exporter en XML une table contenant les champs modèles et de renommer le fichier pour lui donner l'extension accfl. Placez le fichier dans le répertoire templates d'Office et le tour est joué. Le format utilisé vous permettra au chef de projet d'éditer la définition des champs via un simple éditeur de texte, voire même pourquoi pas créer un complément automatisant la tâche.



Beaucoup de fonctionnalités reposent sur ce principe permettant ainsi au développeur de s'approprier l'environnement de développement.

4.2. Le PDF

Sûrement la fonctionnalité qui a fait coulé le plus d'encre : existera, existera pas ... Il y a encore quelques mois, on pouvait lire que la création de document PDF ne serait pas au rendez-vous. Rassurez-vous, il est tout à fait possible de générer des états au format PDF. Toutefois, ce mécanisme n'est pas pris en charge nativement mais nécessite l'installation d'une extension gratuite disponible sur le site Internet de Microsoft.

A travers le tutoriel : Enregistrez vos états au format PDF ([Lien94](#)), je vous invite à faire le point sur les nouvelles possibilités qui s'offre à vous.

5. Conclusion

Comme vous avez pu le lire au travers de ce document, je suis vraiment séduit par cette nouvelle version. Je regrette juste l'orientation "débutant" que prend le produit avec la venue de nouveautés perturbantes pour le professionnel et dangereuses pour le novice (le plus bel exemple : les champs multi-valués).

A noter aussi que la sécurité de niveau utilisateur a disparue (elle fera l'objet d'un prochain article)

Pour résumer, j'invite les développeurs à réaliser au plus vite leurs nouveaux développements sous Microsoft Access 2007, et pourquoi pas migrer l'existant pour bénéficier des avantages du ruban.

Enfin, un dernier mot pour remercier le programme bêta de Microsoft Office 2007 et tous ses interlocuteurs pour la prise en compte rapide des bugs remontés, ainsi que l'équipe Access et Mickaël de www.developpez.com pour leur relecture.

Retrouvez l'article de Christophe Warin en ligne : [Line95](#)

Les derniers tutoriels et articles

Tutoriel sur l'utilisation du client mail Thunderbird

1. Introduction

Cet article a pour but de présenter l'utilisation de thunderbird dans un premier temps pour ceux qui en feront un usage basique et ensuite pour ceux qui veulent aller un peu plus loin.

Cet article se base sur la version 1.5.0.9 pour Windows, bien que le client mail soit aussi disponible pour les plateformes Mac et Linux, tout ce qui est décrit dans cet article ne devrait différer que très peu d'une plateforme à l'autre .

2. Présentation

Thunderbird est le client mail de la fondation Mozilla, ses principaux atouts résident dans le fait que son développement est constant et donc ses mises à jour son plus fréquentes que son concurrent direct de chez Microsoft (Outlook pour ne pas le citer ^^).

Mais aussi que son moteur de rendu html se base sur celui du navigateur Firefox de la même fondation (et toujours en concurrence directe avec son rival de chez Microsoft : Internet explorer) et encore atout majeur : thunderbird est entouré de nombreux modules développés par sa communauté tels des plug-ins, thèmes et autres réjouissances !

2. Utilisation de Base

2.1. Installation

Tout d'abord il faut télécharger le fichier d'installation, si ce n'est pas déjà fait, on peut le trouver ici :

Thunderbird sur Mozilla Europe ([Lien96](#))

Dans le cadre de gauche "Téléchargement", cliquez sur "Windows" (Si vous utilisez Windows évidemment ^^) et téléchargez le fichier d'installation sur votre bureau par exemple . Ensuite un double-clic sur le fichier d'installation et on suit les étapes :

On suit les étapes -> Suivant

On va se faire une installation standard, on se contentera de cliquer sur Suivant jusque la fin ^^ .

2.2. Configurer un compte de courrier

A cet instant Thunderbird se lance, une petite fenêtre s'ouvre pour configurer le compte principal, ici nous allons choisir "Compte courrier électronique"

Ensuite, dans le champs :

- **Votre nom**, on entre le nom qui s'affichera dans la colonne "De" chez le destinataire, soit votre nom .

Exemple "Joris CROZIER" .

- **Adresse de Courrier**, on entre l'adresse électronique associée au compte .E exemple : "jcrozier@dvp.com"

Puis on clique sur Suivant

Ici nous allons configurer la manière dont Thunderbird va aller "chercher et envoyer le courrier" :

On a le choix entre **Pop/Imap**:

POP3/Imap sont des protocoles qui permettent de récupérer les courriers électroniques situés sur un serveur de messagerie électronique .

La principale différence entre les deux :

- Pop3 télécharge l'intégralité du message depuis le serveur jusque votre client mail (le logiciel thunderbird),
- IMAP lui se contente juste de télécharger les entêtes (le nom de l'expéditeur, l'objet et diverses informations concernant l'email), de plus la commande "Suppression" n'est pas prise en compte par ce protocole, les messages sont donc conservés sur le serveur.

De ce fait IMAP protège votre pc des virus plus efficacement car les messages n'étant pas téléchargés sur le client mail, vous ne risquez pas d'ouvrir de pièces jointes compromettantes.

Ceci étant dit à l'heure actuelle, la majorité des utilisateurs ont un antivirus qui vérifie le contenu des emails lorsque ceux ci sont téléchargés sur le client .. ainsi la plupart des virus sont détruits avant leur ouverture...

Nous utilisons donc dans la plupart des cas (et dans notre exemple), le protocole **Pop3**

Dans le champ :

- **Nom du serveur**, on va donc associer le protocole **POP** au domaine de l'adresse email du compte.

Dans mon exemple le domaine est dvp.com, ce qui nous donne pour mon exemple pop.dvp.com (Usuellement on n'utilise plus le 3 de pop3)

Nous laisserons la case "Utiliser la boîte global".

Puis vient la configuration de l'envoi des mails via le protocole smtp:

SMTP signifie **Simple Mail Transfert Protocole** (littéralement « Protocole simple de transfert de courrier », il sert à envoyer ses messages à destinations des serveurs mails de nos destinataires

Dans le champs :

- **Nom du serveur** : nous allons entrer le protocole smtp associé au nom de domaine de l'adresse dans mon

exemple cela donne : smtp.dvp.com

Attention : renseignez vous auprès du service qui "héberge" votre espace mail car certains prestataires n'utilisent pas ces standards, de ce fait les adresses pop et smtp peuvent différer légèrement !

On clique sur **suivant** !

On vous demande maintenant les noms entrant et sortant, ceci correspond à votre login sur l'adresse mail : selon votre "hébergeur" cela peut être le nom avant le @ voir même l'adresse complète !

Dans mon exemple ce pourrait être : **jerozier** ou **jerozier@dvp.com** (remplissez les 2 champs par la même valeur) On clique sur Suivant

Insérez ici le nom du compte, c'est ce nom qui s'affichera dans la liste de vos comptes !

On clique sur **Suivant** !

Voilà c'était pas si compliqué au final, non ?

On va cocher la case "Télécharger les messages maintenant" et cliquer sur **Terminer**

2.3. Recevoir son courrier

Une petite fenêtre s'ouvre et vous demande le mot de passe du compte pour aller chercher votre courrier

Vous pouvez cocher la case "Utiliser le gestionnaire de mot de passe pour retenir ce mot de passe"

Vous verrez c'est aussi très pratique on évite de retaper le mot de passe à chaque relève du courrier

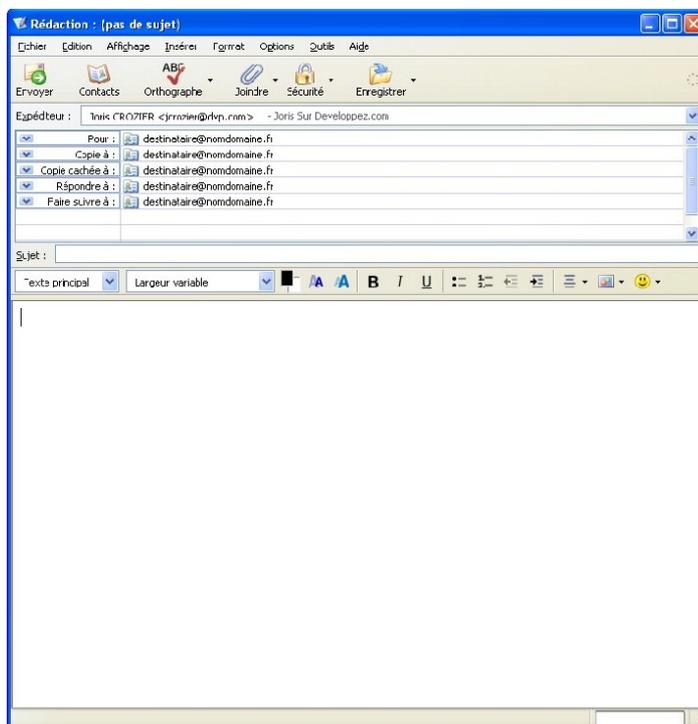
Evidement on va taper le mot de passe de l'adresse mail dans le champ prévu à cet effet !

- Corbeille : Les messages supprimés
2. La fenêtre des entêtes des messages du dossier choisis dans la fenêtre 1
3. La fenêtre d'affichage du corps du message choisis dans la fenêtre 2

Pour relever le courrier de ses comptes mail, un petit clic sur Relever et les messages sont téléchargés dans Thunderbird.

2.4. Ecrire un mail

Sur notre barre d'outils on va cliquer sur Ecrire, une fenêtre s'ouvre

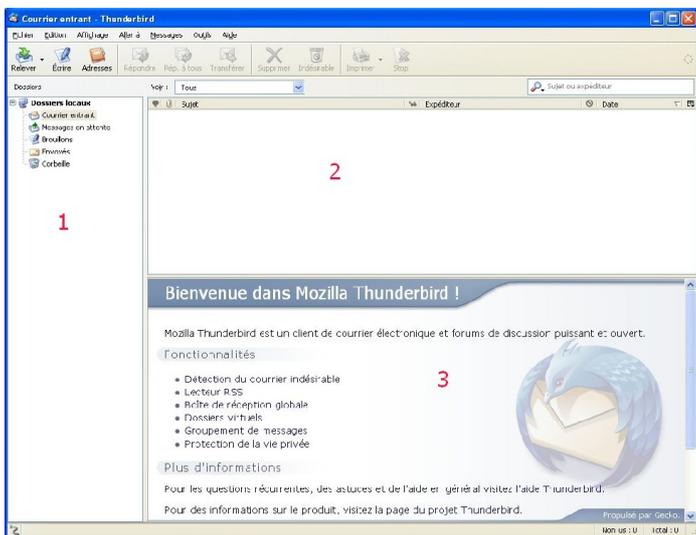


Nous voici face à notre fenêtre ! La barre d'outils comprends plusieurs boutons :

- **Envoyer** : Envoi notre mail
- **Contacts** : Ouvre le carnet d'adresse (Nous en reparlerons plus tard)
- **Orthographe** : Permet de choisir la langue du dictionnaire utilisé pour corriger les fautes dans le messages (Nous reparlerons de l'ajout de dictionnaire plus tard)
- **Joindre** : Permet de joindre un fichier à notre email. nb : attention vous ne pouvez pas joindre des fichiers trop volumineux à vos emails, la plupart des hébergeurs préconisent de ne pas dépasser les 2 Mo, mais ceci peut varier selon les prestataires ! **Sécurité** : Pour chiffrer ses emails (Nous en reparlerons plus tard)
- **Enregistrer** : Permet d'enregistrer son mail, soit comme Fichier au format texte, soit comme Brouillon, soit comme Modèle, permettant dans le troisième cas de se servir de l'email sans le retaper à chaque fois !

Le champs **Expéditeur** : L'expéditeur du mail (a savoir : vous même lol), ce champs représente une liste déroulante car, il y a autant d'expéditeurs que vous avez de comptes configurés dans Thunderbird Vient ensuite un cadre réservé au destinataires, on peut y voir un tableau colonne, la première représentant le mode d'envoi au destinataire, la seconde est réservée à l'adresse de celui ci !

Le mode d'envoi :



Voilà notre écran principal, il se compose de 3 parties :

1. L'arborescence des dossier :
 - Courrier entrant : Le dossier où vos mails arrivent par défaut
 - Messages en attente : Les mails en attente d'envoi sont placés ici
 - Brouillons : Les messages enregistrés comme brouillons
 - Envoyés : Les messages envoyés

- **Pour** : Le mail est transmis à cette adresse simplement, c'est le destinataire principal
- **Copie à** : Le mail est envoyé au(x) destinataire(s) principal, et une copie est adressée au(x) destinataire(s) de cette ligne **NB** : Le destinataire en copie peut voir l'adresse du destinataire principal dans les entêtes du mail
- **Copie cachée** : Le mail est envoyé au(x) destinataire(s) principal, et une copie est adressée au(x) destinataire(s) de cette ligne **NB** : Le destinataire en copie ne peut pas voir l'adresse du destinataire principal dans les entêtes du mail
- **Répondre à** : Cette option permet de spécifier une adresse mail de réponse différente de celle utilisée pour l'envoi du mail, ainsi le destinataire en cliquant sur **Répondre** enverra la réponse à l'adresse spécifiée
- **Faire suivre à** : Une copie du mail est transférée, à l'adresse spécifiée

Le champs **Sujet**, désigne donc l'objet principal, ce sujet s'affichera chez l'expéditeur dans l'entête du message (Dans nos emails reçu, dans la fenêtre 2 de notre page principale)

Le cadre d'édition du message : Ici, on tape le corps du message, par défaut les dictionnaires sont activés, les mots jugés mal orthographiés seront soulignés en rouge ! Un clic droit dévoilera une proposition d'orthographe du dictionnaire !

Par défaut les messages sont envoyés au format texte, La barre d'outils entre, le champs sujet et le cadre d'édition du message, permet tout comme dans un logiciel de traitement de texte, de mettre en forme son message, et d'y insérer de images .

Dans ce cas le mail sera envoyé en html .

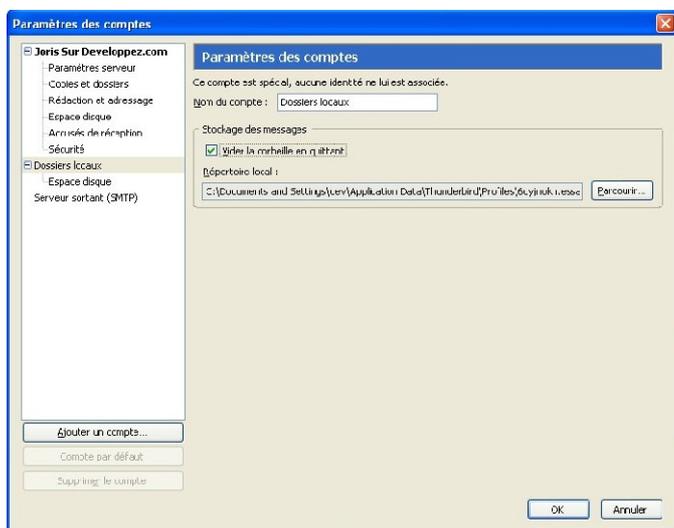
Une fois le mail terminé, les destinataires choisis et le sujet défini (vous pouvez laisser le sujet vide mais Thunderbird vous demandera de confirmer ce choix avant l'envoi du mail), vous pouvez envoyer votre message en cliquant sur **Envoyer**.

2.5. Supprimer ses mails

Lorsque vous supprimez un mail, il va directement dans le dossier Corbeille, mais il n'est pas pour autant supprimé, il est encore sur votre disque dur et prend de la place !

Plusieurs solutions pour vider définitivement le dossier Corbeille :

- Un clic droit sur le dossier Corbeille, puis **Vider la Corbeille**
- Une méthode automatique plus pratique : Dans le menu **Outils -> Paramètres des comptes ->** puis dans le menu **Dossier locaux cf.. Image :**



On coche la case **Vider la corbeille en quittant**, les messages seront supprimés chaque fois que Thunderbird sera fermé .

2.5. Quelques Options dites 'Avancées'

Dans le menu Paramètres des comptes, sur la ligne Paramètres Serveurs de vos comptes emails, vous pouvez modifier plusieurs options :

- **Vérifier le courrier au lancement** : Le courrier est relevé à chaque lancement de Thunderbird
- **Vérifier les nouveaux messages toutes les** : Spécifier un délais en minutes et Thunderbird vérifiera les boites mails à intervalles réguliers (nécessite que Thunderbird soit allumé)
- **Télécharger Automatiquement les nouveaux messages** : Les messages sont téléchargés en complet depuis le serveur
- **Récupérez les entêtes uniquement** : Seuls les entêtes sont téléchargés augmentant la vitesse de récupération des mails et le risque de télécharger un virus, en cliquant sur l'entête du mail lors de la lecture, Thunderbird vous demande si vous voulez télécharger le reste du message (donc le corps)
- **Laisser les messages sur le serveur** : Permet de laisser une copie du message sur le serveur, idéal lorsque vous récupérez vos mails sur le Thunderbird d'un autre poste, vous pouvez à nouveaux les récupérer chez vous par la suite !

2.7. Filtre AntiSpam

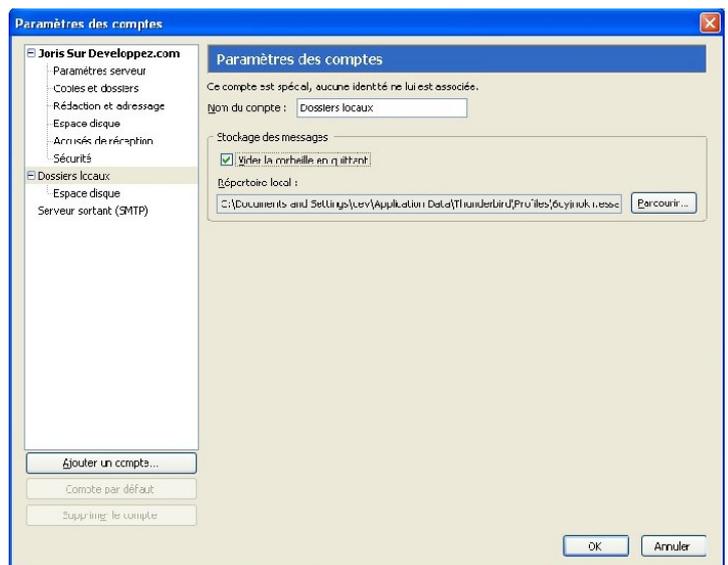
Avec Thunderbird, on peut efficacement trier ces mails lors du téléchargement de ceux et sans aucune action de notre part !

Il est capable de reconnaître le Spam (et en cas d'échec apprendre et se perfectionner), rendant ainsi plus agréable la relève du courrier !

On va donc l'activer : cliquez sur **Outils -> Gestionnaire des indésirables**

Une fenêtre vous expliquant le gestionnaire s'affiche : un clic sur **OK** !

Puis la fenêtre principale s'affiche :



Nous allons donc faire quelques petits réglages rapides :

Configurer les paramètres du Courrier Indésirable pour : On laisse **Dossier locaux**

Onglet Paramètres :

- On coche la case **Ne pas marquer les messages**

indésirables **si l'expéditeur est dans : Adresses Personnelles**, ce qui aura pour effet de faire confiance à vos adresses contenues dans le carnet d'adresse .

- On laisse la case décochée **Se fier aux entêtes de courrier indésirables placées par**, utilisée en parallèle de SpamAssassin ou SpamPal mais vous verrez Thunderbird se débrouille très bien tout seul à la longue !
- On coche **Déplacer les messages entrant Identifiés comme Indésirables vers** puis on coche **Dossier Indésirable** sur et on laisse Dossier Locaux , les messages seront donc redirigés dans ce dossier si ils sont reconnus comme Indésirables !

Prenez l'habitude surtout au début de vérifier le contenu du dossier Indésirables car dans un premier temps certains mails voulus risquent de finir ici !

Cocher la case **Supprimer [...] au bout de 14 jours** (ou moins) en prenant garde à ma remarque juste au dessus, sinon ne cochez pas cette case mais videz votre dossier indésirables régulièrement !

Enfin cochez la case **Assainir l'Affichage des [...]**, qui aura pour effet de n'afficher qu'en partie les messages indésirables et vous protèges d'éventuels scripts inclus dans les mails

Onglet **Filtre Adaptatif**:

Cochez la case **Activer la détection adaptative [...]**, ceci pour que Thunderbird apprenne à reconnaître le Spam et en laisse passer de moins en moins

Justement comment différencier les mails manuellement ?

Tout simplement en cliquant sur votre mail : dans la barre d'outils de la fenêtre principale Thunderbird se trouve un bouton Indésirables, un clic sur ce bouton et le mail sélectionné est automatiquement géré comme Spam (et nourrit la base informative de Thunderbird)

A l'inverse si un message est considéré comme Spam à tort, on procède de la même manière sauf que cette fois ci le bouton Indésirables à pris le nom d'**Acceptable** !

3. Utilisation Avancée

3.1. Utilisation du Carnet d'adresse

Nous allons jeter un petit coup d'oeil au carnet d'adresse : Outils -> Carnet d'adresse

Avec la possibilité d'ajouter des fiches : une fiche correspond à une personne, et des listes

Les listes sont un moyens très pratiques d'envoyer un mail à un groupe de personne sans avoir à entrer toutes leurs adresses une par une

Pour ajouter une nouvelle liste, on choisit :

- Un nom pour notre liste un pseudo Eventuellement une description
- Et enfin on entre les Adresses des destinataires inclus dans notre liste

Ensuite lors de la création d'un mail, dans le champs destinataire : il suffira d'entrer le nom ou le pseudo de la liste, Thunderbird se charge de faire le lien !

3.2. Classer vos mails

Il est possible de classer ces emails automatiquement lors de la relève du courrier, nous allons voir comment :

Dans un premier temps nous allons créer un dossier Documents dans les Dossiers locaux (clic-droit sur dossier locaux puis Nouveau Dossier)

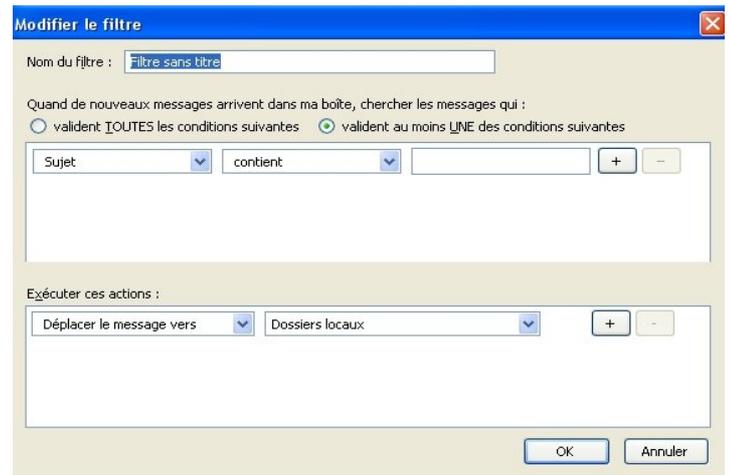
Maintenant on va créer des règles pour filtrer nos mails : cliquez sur Outils -> Filtre des messages

Notre fenêtre s'affiche :



Dans la boîte de dialogue nous avons le choix entre chacun des comptes configurés et Dossier Locaux (Donc tout les comptes confondus)

Et nous allons cliquer sur Nouveau



On entre tout abord un nom à notre règle

Il faut choisir si notre mail doit remplir TOUTES les conditions que l'on va énumérer ou AU MOINS UNE

Le premier cadre est utilisé pour les "Conditions" que le mail doit remplir

Le second cadre est utilisé pour les "Actions" à effectuer sur le dit mail si il remplit les conditions précédentes

Exemple :

Dans le cadre des Conditions :

- Première liste déroulante : Nous laisserons Sujet
- Seconde liste Déroulante : Nous laisserons Contient
- Puis dans le champs nous allons mettre : Documents

Dans le cadre des Actions :

- Première liste déroulante : Nous laisseront Déplacer le message vers :
- Seconde liste Déroulante : Nous allons aller chercher notre dossier Documents

On valide ! A chaque fois que le mot Documents sera dans le sujet le mail sera redirigé dans le dossier Documents (Simple comme bonjour non ?)

Le + et le - à coté de chaque ligne sert à ajouter une Condition/Action

3.3. Utilisation du Dictionnaire Orthographique

Chaque mail que l'on créé peut être vérifié au niveau de l'orthographe, pour se faire il faut aller chercher le dictionnaire français et l'activer :

Nous allons aller dans **Outils -> Options -> Rédaction -> Onglet Orthographe**

On va cocher la case : **Activer la vérification pendant la saisie**, et chaque mots mal orthographiés sera souligné en rouge dans notre mail

Puis on va cliquer sur **Télécharger d'autres dictionnaires** (Si le français n'est pas présent dans la liste déroulante ce qui est mon cas)

Vous allez être redirigé sur :

- Le site de Mozilla ([Lien97](#))

Puis on va aller chercher le Français (et tout ceux qui vous intéresse si vous écrivez en polonais rien ne vous empêche de télécharger le dictionnaire de cette langue)

On l'enregistre sur le bureau par exemple

Pour installer le dictionnaire, nous allons ouvrir le gestionnaire d'extension : **Outils -> Extensions -> Installer** et on va aller chercher le ou les dictionnaire(s) sur le bureau

On valide et le tour est joué, vous pouvez retourner dans les options et sélectionner votre dictionnaire dans la liste déroulante !

3.7. Chrome

Qu'est-ce que Chrome ?

Chrome est une interface utilisateur implantée dans vos fenêtres et hors de l'espace de travail en gros ce qui se rapporte aux barres d'outils, barres d'états, barres de progressions, barres de titres et bien d'autres.

Les Fournisseurs de Chrome :

- **Contenu** : Ce sont des fichiers expliquant le contenu des interfaces chrome, généralement des fichiers .xul
On peut aussi y trouver des fichiers contenant du javascript généralement .xbl
- **Local** : Un repertoire local contient toutes les informations d'une interface chrome indépendant des sources de l'application elle même comme par exemple les extensions, les traductions du soft etc ...
Ces fichiers portent souvent l'extension DTD

- **Skin** : Comme son nom l'indique, ce fournisseur est responsable de l'apparence graphique des interfaces chrome, généralement des fichiers .css et des images .

Le registre de Chrome :

Le moteur de Gecko maintient un service appelé "Le registre de chrome" qui fournit les liaisons entre les elements chrome de l'application et leur localisation physique sur le disque dur.

Concretement Chrome est donc un ensemble d'elements fournit par Mozilla pour normaliser la personnalisation de ses suites logicielles comme ThunerBird, FireFox, SeaMonkey et autres

Nous en resterons volontairement à ce stade sur Chrome pour notre Guide, il fera peut etre l'objet d'un tutoriel complet dans le futur ...

3.8. Thèmes et Extensions

Vous pouvez changer l'apparence de votre Thunderbird tres simplement en allant chercher des thèmes :

Thèmes pour Thunderbird ([Lien98](#)) : Pour les installer : menu Outils -> Themes -> Installer, et il vous reste plus qu'à aller chercher le fichier de votre thème téléchargé auparavant !

Vous pouvez aussi ajouter des fonctions (utiles ou non lol) à votre Thunderbird via les Extensions ([Lien99](#))

Pour les installer : menu Outils -> Extensions, et il vous reste plus qu'à aller chercher le fichier de votre extensions téléchargé auparavant !

Je vous laisse le soin de decouvrir et choisir Vos Thèmes/Extensions à votre guise !

5. Conclusion

Avec un developement sans cesse en mouvement, une communauté gigantesque, des thèmes par milliers, des extensions pour toutes les utilisations possibles Thunderbird est un des Tenors des clients mail !

Son utilisation tres proche de l'utilisateur lambda, séduit, son évolution permanente ne laisse pas de marbre et sa personnalisation en fait un argument de choix !

En esperant avoir pu vous séduire, et vous apprendre d'avantages de choses sur ce client mail !

Je vous tire ma révérence !

Retrouvez l'article complet de Joris Crozier en ligne : [Lien100](#)

Les derniers tutoriels et articles

Utiliser l'OTA (Open Tools API) de Borland Developer Studio 2006

Dans ce tutoriel, je vais vous montrer comment démarrer simplement un projet vous permettant de créer vos propres extensions à l'environnement de développement BDS.

L'architecture permettant de le faire s'appelle OTA (Open Tools API). BDS 2006 fournit deux OTA. Autourd'hui, nous n'utiliserons que l'OTA accessible en .NET. On pourrait par conséquent écrire ces extensions aussi bien en Delphi pour .NET qu'en C#. Nous ne ferons que du langage Delphi pour .NET dans ce tutoriel.

Ce document regroupe un ensemble d'informations que j'ai collectées à travers de nombreux sites web et j'essaye d'être le plus didactique possible en expliquant chacune des étapes à effectuer pour ne pas rencontrer de problème.

1. Affichage d'une boîte de dialogue de message au démarrage de l'EDI

Lancer BDS 2006, Delphi pour Microsoft .NET Framework.

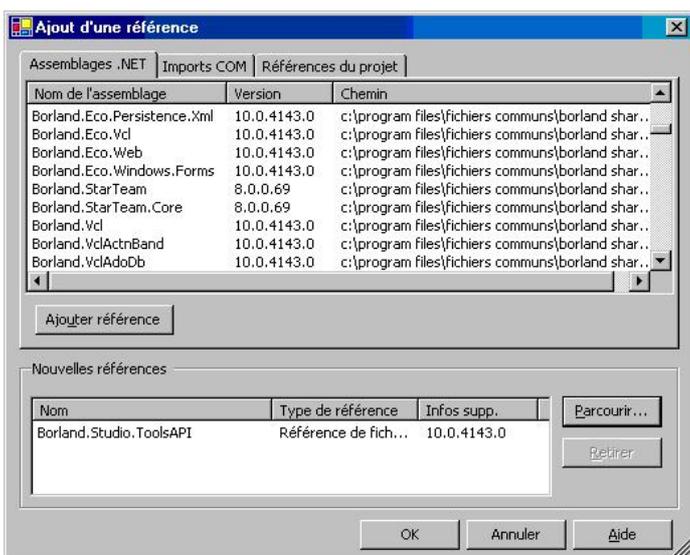
Créer une nouvelle bibliothèque, en cliquant sur :

- Menu Fichier/Nouveau/Autre...
- Projets Delphi pour .NET
- Bibliothèque

Sauvegarder le projet (pas de parenthèse dans le chemin)

Ajouter l'assemblage (assembly) pour accéder à l'OTA :

- Menu Projet/Ajouter une référence...
- Choisir l'assemblage : Borland.Studio.ToolsAPI
 - Si vous ne le trouvez pas, cliquez sur le bouton Parcourir...
 - Choisissez le fichier Borland.Studio.ToolsAPI.dll du répertoire bin de votre installation de BDS.
- Cliquez sur le bouton OK.



Modifions le code pour que l'extension que nous avons créée fasse quelque chose :

- Ajoutons une nouvelle avec une bien spécifique :

```
//[assembly: Guid('')]
//[assembly: TypeLibVersion(1, 0)]

type
  TMyFirstAddIn = class
  public
    class procedure IDERegister; static;
  end;

{ TMyFirstAddIn }

class procedure TMyFirstAddIn.IDERegister;
begin
  MessageBox.Show('Hello World !');
end;

begin
end.
```

- Les assemblages .NET publient leur structure ce qui permet aux autres applications de facilement découvrir les fonctions et procédures qu'ils contiennent. BDS va se livrer à l'introspection de ses extensions pour en rechercher une procédure IDERegister, puis l'exécuter lorsqu'il l'aura trouvée.
- La classe MessageBox utilisée est contenue dans l'espace de noms : System.Windows.Forms (que l'on doit donc ajouter au uses de la bibliothèque).
- On peut ensuite sauvegarder son code et compiler son projet.

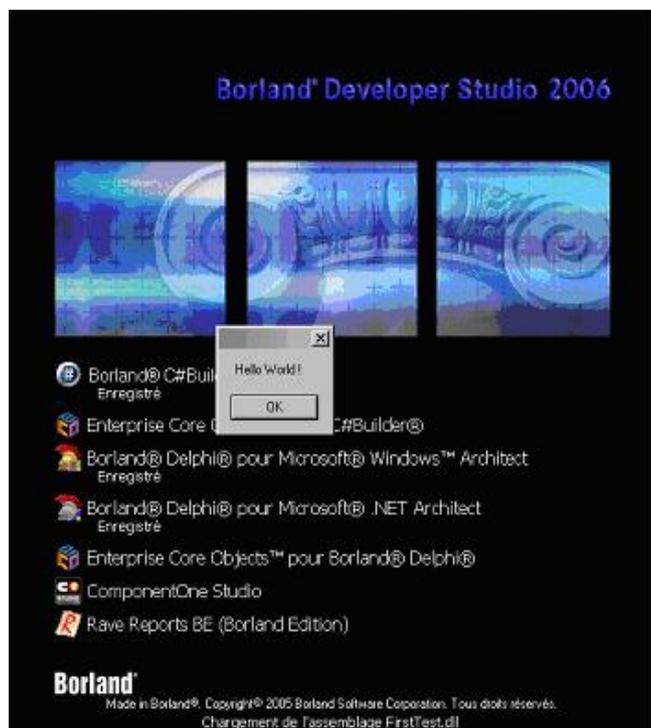
Maintenant, il faut que BDS charge cette nouvelle extension lorsqu'il se lance. Pour cela, il faut ajouter une nouvelle clé dans la base de registre :

- Lancez l'Editeur du Registre :
 - Menu Démarrer/Exécuter... (ou touche Windows + R)
 - Tapez regedit puis cliquez sur OK
- Parcourez l'arborescence pour aller sur la clé : HKEY_CURRENT_USER\Software\Borland\BDS\4.0\Known IDE Assemblies (4.0 correspond à BDS 2006, 3.0 pour 2005, ...)
- Créez une nouvelle valeur chaîne et donnez-lui comme nom le chemin complet pour accéder à la DLL que nous venons de compiler. La valeur de la donnée par elle-même est sans importance mais ne doit pas être vide.

Quittez BDS et relancez-le : Lors du chargement, le message

```
[assembly: ComVisible(False)]
```

Hello World ! apparaît...



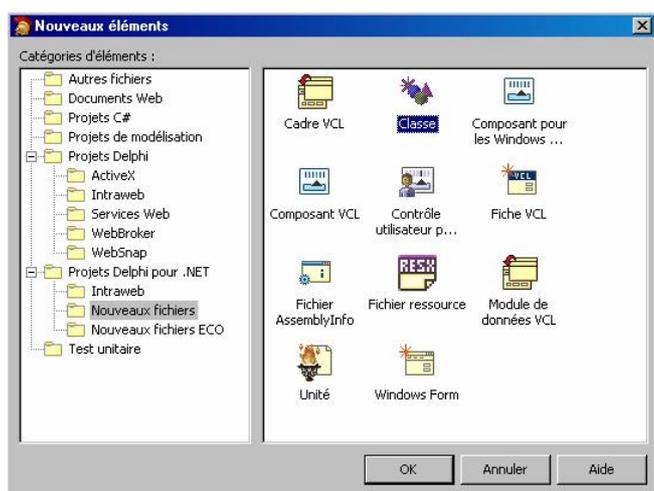
Nous avons ainsi réussi à créer notre première extension. Maintenant, nous allons essayer d'en faire une à peine plus complexe : La même mais où le Hello World sera affiché à partir d'un menu.

2. Créer un nouvel élément de menu dans l'EDI

Nous allons donc créer une nouvelle bibliothèque comme nous l'avons précédemment fait.

Nous lui ajoutons également la référence à l'assemblage Borland.Studio.ToolsAPI.

Puis nous créons une nouvelle classe.



Cette classe va contenir notre nouvelle extension. Elle hérite directement d'un TObject et implémente l'interface de IOTA IOTAMenuWizard qui impose des propriétés permettant de récupérer le texte du menu, son nom, ...

Pour cela, on va modifier le code généré par BDS.

Pour commencer, on va utiliser les espaces de noms suivants :

```
uses
```

```
Borland.Studio.ToolsAPI,  
System.Windows.Forms;
```

Puis faire la déclaration ainsi :

```
type
```

```
TExtensionSimpleOTA = class(TObject, IOTAMenuWizard)  
strict private  
    procedure CreateWizardService;  
private  
    { Déclarations privées }  
public  
    constructor Create;  
    procedure Execute;  
    procedure Destroyed;  
  
    class procedure IDERegister; static;  
  
    function FGetIDString : String;  
    function FGetName : String;  
    function FGetMenuText : String;  
    function FGetEnabled : Boolean;  
    function FGetChecked : Boolean;
```

```
    property IDString : String read FGetIDString;  
    property Name : String read FGetName;  
    property MenuText : String read FGetMenuText;  
    property Enabled : Boolean read FGetEnabled;  
    property Checked : Boolean read FGetChecked;  
end;
```

```
implementation
```

On remarque que l'on retrouve comme dans le premier exemple la procédure qui permet à BDS d'enregistrer l'extension, IDERegister. Celle-ci va créer une instance de l'extension.

Dans le constructor de l'extension, on va faire appel à la procédure qui permet d'ajouter un nouvel expert à l'environnement, qui dans notre cas, sera un menu supplémentaire.

La procédure Execute sera quant à elle appelée lorsque l'on cliquera sur le menu et devra donc afficher le message Hello World.

L'implémentation ressemblera donc à ceci :

```
implementation
```

```
class procedure TExtensionSimpleOTA.IDERegister;  
begin  
    TExtensionSimpleOTA.Create;  
end;
```

```
constructor TExtensionSimpleOTA.Create;  
begin  
    inherited Create;  
    // TODO: Ajouter ici le code du constructeur  
    CreateWizardService;  
end;
```

```
procedure TExtensionSimpleOTA.CreateWizardService;  
var  
    WizardService: IOTAWizardService;  
begin  
    WizardService :=  
    BorlandIDE.GetService(typeof(IOTAWizardService)) as  
    IOTAWizardService;  
    WizardService.AddWizard(Self);
```

```

end;

procedure TExtensionSimpleOTA.Execute;
begin
    MessageBox.Show('Hello World à partir du menu que
l'on a créé !');
end;

procedure TExtensionSimpleOTA.Destroyed;
begin
    // Rien à faire : Le ramasse-miette s'en chargera...
end;

function TExtensionSimpleOTA.FGetChecked: Boolean;
begin
    Result := False;
end;

function TExtensionSimpleOTA.FGetEnabled: Boolean;
begin
    Result := True;
end;

function TExtensionSimpleOTA.FGetIDString: String;
begin
    Result := 'ExtensionSimpleOTA';
end;

function TExtensionSimpleOTA.FGetMenuText: String;
begin
    Result := 'Exécuter ma nouvelle extension !';
end;

function TExtensionSimpleOTA.FGetName: String;
begin
    Result := 'Extension simple via OTA';
end;

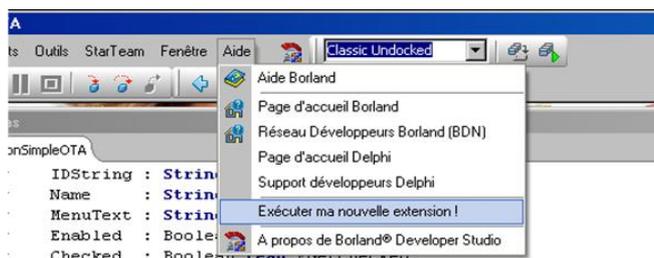
end.

```

Ensuite, il suffit de sauvegarder et de compiler son projet.

Puis on ajoute dans la base de registre l'appel à notre nouvelle extension (DLL).

On obtient ainsi un nouveau menu :



Ce menu nous affiche effectivement le message que nous avons codé.

Maintenant, lors du chargement de BDS, vous avez remarqué que certaines extensions affichent un logo lorsqu'elles-mêmes sont chargées. Nous allons maintenant voir comment procéder pour faire la même chose.

3. Affichage de l'extension dans l'écran de démarrage de l'EDI (Splash screen)

Tout d'abord, nous allons avoir besoin d'une ressource image. Pour la créer, nous allons utiliser l'application Démo fournie avec

BDS, ResXBuilder qui se trouve dans le répertoire : <BDS>\Demos\Delphi.Net\WinForms\ResXBuilder\

L'image doit faire une taille de 24x24.

Une fois le projet ResXBuilder ouvert, nous créons un nouveau fichier :

- Menu File/New

Puis nous ajoutons une ressource de type Bitmap :

- Menu Resources/Add/Bitmap...
- Choisir le fichier voulu et cliquer sur Ouvrir

Puis nous sauvegardons notre fichier :

- Menu File/Save

Ensuite, nous ouvrons notre projet précédent pour le modifier afin qu'il utilise notre logo fraîchement créé.

Nous ajoutons le logo à notre projet :

- Menu Projet/Ajouter au projet... (ou Maj + F11)
- Nous choisissons le type de fichier Ressources (*.resx)
- Nous cliquons sur le fichier créé avec le projet ResXBuilder et sur le bouton Ouvrir.

Après, nous modifions notre code pour qu'il utilise cette nouvelle ressource, en ajoutant une nouvelle procédure qui va être appelée à partir du constructeur. Comme cette procédure va utiliser des espaces de noms particuliers, nous allons tout d'abord les ajouter :

```

uses
    Borland.Studio.ToolsAPI,
    System.Windows.Forms,
    System.Drawing,
    System.Resources,
    System.Reflection;

```

Puis, nous ajoutons la déclaration de la nouvelle procédure :

```

strict private
    procedure CreateWizardService;
    procedure CreateSplashImage;
private

```

L'appel de cette procédure dans le constructeur :

```

constructor TExtensionSimpleOTA.Create;
begin
    inherited Create;
    // TODO: Ajouter ici le code du constructeur
    CreateSplashImage;
    CreateWizardService;
end;

```

Et enfin, l'implémentation de la procédure :

```

procedure TExtensionSimpleOTA.CreateSplashImage;
var
    bmp : System.Drawing.Bitmap;
    SplashScreenService : IOTASplashScreenService;
    lAssembly : System.Reflection.Assembly;
    leResourceManager : System.Resources.ResourceManager;
begin
    // Récupération du service pour la splash
    SplashScreenService :=
    BorlandIDE.GetService(typeof(IOTASplashScreenService))
    as IOTASplashScreenService;

```

```

// Récupération de l'assemblage courant
lAssembly := GetType().Assembly;

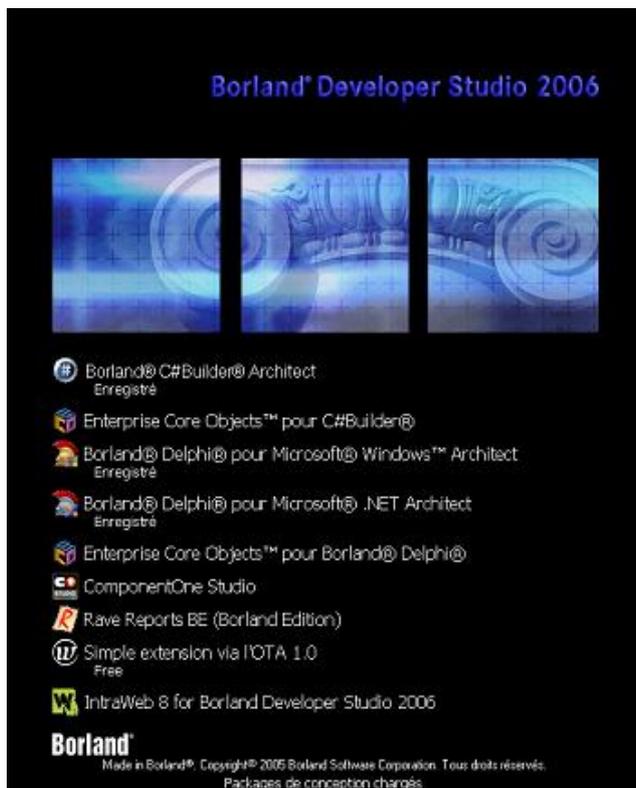
// Récupération du gestionnaire de ressources
leResourceManager :=
System.Resources.ResourceManager.Create('whiler',
lAssembly);

// Récupération de l'image que nous avons créé
(attention, sensible à la casse)
bmp :=
System.Drawing.Bitmap(leResourceManager.GetObject('whiler.bmp'));

// Ajout de l'image
SplashScreenService.AddPluginBitmap('Simple extension
via l''OTA', bmp.GetHbitmap, False, 'Free', '1.0');
end;

```

On remarque sur la capture ci-dessous l'emplacement des textes des différents paramètres :



4. Affichage de l'extension dans la fenêtre A propos de...

L'architecture ouverte que nous sommes en train d'utiliser a été conçue pour permettre à des éditeurs tiers de créer des extensions. De la même façon que nous avons ajouté des informations sur l'écran de démarrage (splash screen), nous allons maintenant ajouter des informations dans la fenêtre A propos de... de BDS.

Pour cela nous allons faire presque la même chose que pour la splash. L'image utilisée peut faire une taille de 32x32.

Nous créons une nouvelle procédure CreateAboutInfos que nous appelons à partir du constructor et que nous implémentons :

Déclaration de la nouvelle procédure :

```
strict private
```

```

procedure CreateWizardService;
procedure CreateSplashImage;
procedure CreateAboutInfos;
private

```

L'appel de cette procédure dans le constructor :

```

constructor TExtensionSimpleOTA.Create;
begin
    inherited Create;
    // TODO: Ajouter ici le code du constructeur
    CreateSplashImage;
    CreateAboutInfos;
    CreateWizardService;
end;

```

Et enfin, l'implémentation de la procédure :

```

procedure TExtensionSimpleOTA.CreateAboutInfos;
var
    bmp                : System.Drawing.Bitmap;
    AboutBoxService    : IOTAAboutBoxService;
    lAssembly          : System.Reflection.Assembly;
    leResourceManager  :
System.Resources.ResourceManager;
begin
    // Récupération du service pour le About
    AboutBoxService:=
BorlandIDE.GetService(typeof(IOTAAboutBoxService)) as
IOTAAboutBoxService;

    // Récupération de l'assemblage courant
    lAssembly := GetType().Assembly;

    // Récupération du gestionnaire de ressources
    leResourceManager :=
System.Resources.ResourceManager.Create('whiler',
lAssembly);

    // Récupération de l'image que nous avons créé
(attention, sensible à la casse)
    bmp :=
System.Drawing.Bitmap(leResourceManager.GetObject('about.bmp'));
    // On met le fond de l'image transparent
    bmp.MakeTransparent;

    // Ajout de l'image
    AboutBoxService.AddPluginInfo('Extension via l''OTA',
'Un exemple simple', bmp.GetHbitmap, False, 'Free',
'1.0');
end;

```

On obtient ainsi notre extension dans la liste des produits installés dans la fenêtre « à propos ».

5. Conclusion

J'espère avoir été suffisamment clair dans ce tutoriel et qu'il vous aura permis de plus facilement aborder l'architecture qui vous permettra ainsi de créer vos propres extensions.

Si vous obtenez cette erreur lors de la compilation de votre projet, Erreur du Linker pendant l'émission des métadonnées (E2328), c'est que votre extension a été montée dans BDS et que celui-ci ne peut par conséquent pas recréer la DLL lors de la compilation.

Retrouvez l'article de Whiler en entier en ligne : [Lien101](#)

Défi – Jouez au puissance 4

Le troisième défi proposé par l'équipe Delphi est la création du jeu du puissance 4 !

Pour ceux qui ne le sauraient pas encore, l'équipe Delphi de developpez.com organise des défis pour les membres du forum.

Le troisième défi proposé par l'équipe Delphi a été la création du jeu du puissance 4 ! Les participations ont été nombreuses et se sont faites dans la joie et la bonne humeur qui s'impose !

Vous pouvez d'ores et déjà consulter les résultats de ce défi ([Lien118](#)), mais peut-être aussi jeter un oeil au défi en cours ([Lien119](#)) et y participer !

1. Sujet du défi

Le troisième défi proposé par l'équipe Delphi est la création du jeu du puissance 4 !

Il s'agit de réaliser le jeu selon les règles du puissance 4, afin que 2 joueurs sur le même ordinateur puissent s'affronter.

1.1. Les objectifs du défi

Le logiciel doit comporter les éléments suivants :

1. Créer l'interface utilisateur du jeu
2. Respecter les règles du Puissance 4:
 - Etablir un plateau de 7 colonnes sur 6 lignes
 - Chaque joueur possède 21 jetons dans sa couleur
 - Chaque jeton joué est lâché au sommet d'une colonne et prend sa place dans la première cellule libre en partant du bas
 - Le 1er joueur alignant 4 jetons dans sa couleur (horizontal, vertical ou diagonal) a gagné, ou si les 42 jetons sont joués, la partie est nulle
3. Permettre à 2 joueurs sur le même ordinateur, de jouer l'un contre l'autre
4. Laissez libre court à votre imagination et proposez vos propres fonctionnalités.

Les participants doivent respecter les règles du défi, et le déroulement du défi et plus précisément que "l'utilisation de composantes ou bibliothèques autres que celles fournies en standard par Borland sont interdites, qu'elles soient commerciales, freewares, open-source etc. ..."

2. Le défi: topic

Comme pour tous les défis, ce défi s'est déroulé sur le forum DELPHI.

Vous pouvez désormais retrouver sur cette page ([Lien103](#)) l'archive du sujet concernant ce défi.

3. La solution du défieur

TicTacToe, selon les règles du défi, a réussi à mettre au point un tel logiciel avant que le défi ne soit lancé sur le forum.

3.1. Introduction

Le programme s'articule autour des 2 classes TPlateau et TJoueur.

3.1.1. TJoueur

La classe TJoueur garde les informations relatives à chaque joueur.

- **Plateau**: le plateau d'appartenance
- **SNom**: son nom.
- **ICamp**: son camp. (actuellement 1=rouge, 2=jaune)

- **IJetonsRestants**: son nombre de jetons restants au cours d'une partie

2 méthodes également, TJoueur.**Init** et TJoueur.**Jouer**, permettent de faire interagir le joueur au cours de la partie.

3.1.2. TPlateau

La classe TPlateau garde les informations du plateau, principalement:

- **Joueur[x]**: Les joueurs de classe T**Joueur**, indicés de 1 à 2.
- **Grille[col,lig]**: grille virtuelle aux dimensions du puissance 4 (7 colonnes par 6 lignes)
Les valeurs des cellules sont: 0 si la cellule est vide, 1 ou 2 pour spécifier le camp du jeton.
- **DernierJeton**: Les coordonnées du dernier jeton joué.
- **ICurseur**: Curseur en cours du laché de jeton.
- **IJoueur**: Joueur actuellement en cours (1 ou 2). Vaut 0 si aucune partie n'est pas en cours.

Ainsi que des informations pour l'affichage:

- **BmpFond**, **BmpJeu** sont les images de fonds et de grille de puissance 4
- **BmpDoubleFond** est l'image de fond, pour gérer un affichage correct pendant les animations des jetons
- **BmpPion[x]** sont les images des jetons

TPlateau intègre également ses propres méthodes pour le faire vivre, principalement les méthodes:

- **NouveauJeu**: Initialise une partie et ses joueurs.
- **Jouer**: Lache un jeton selon l'état courant du jeu (à la position ICurseur) et gère l'analyse des jetons alignés.
- **Fin**: Partie terminée, affiche le message de match nul ou gain.

Les autres méthodes sont des outils, citées et expliquées par la suite.

3.2. Algorithme général

Etapes principales pendant le déroulement du jeu

1. Le plateau est initialisé avec **IJoueur** = x (x = 1 par exemple si le joueur N°1 commence)
2. La propriété **ICurseur** est mise à jour en temps réel, lorsque la souris se déplace sur le plateau.
3. Le clic-souris sur le plateau, déclenche le laché de jeton par **Plateau.Jouer**.
 - le joueur courant **IJoueur** lache un jeton dans la colonne courante **ICurseur**.
 - **TPlateau.CaseVide** indique quelle est la cellule libre à remplir, l'algorithme est suffisamment simple à comprendre en lisant le code.
4. L'analyse de gain / nul est lancé avec la méthode **Plateau.ResultatDernierJeton**, expliqué ci-dessous.
5. Si la partie continue, le programme boucle en 2.

TPlateau.CaseVide permet de savoir, en fonction de la colonne du laché de jeton, quelle est la case finale du jeton.

La recherche d'une case vide est faite par un parcours séquentiel dans la colonne désignée, de haut en bas, à la recherche d'une cellule déjà remplie par un jeton.

Analyse du dernier jeton joué

Le programme analyse uniquement à la périphérie du dernier jeton joué, l'analyse ne porte pas sur l'ensemble de la grille.

J'ai choisi cette méthode car un minimum de jetons seront soumis à des tests, le code est également assez compact.

De plus, il est permis d'imaginer de faire évoluer le plateau en nombre de cellules et/ou en nombre de dimensions sans changer l'algorithme.

Le vocabulaire utilisé est: **SENS** = axe non orienté, **DIRECTION** = axe orienté.

L'analyse repose sur le "comptage" des jetons dans les 8 **directions** de cellules identiques au dernier jeton joué.

Puis ensuite au comptage des jetons pour les 4 **sens**.

Pour effectuer ce comptage dans les 4 sens, la somme des compteurs de directions 2 à 2 opposées et effectuée + 1 (le jeton central).

Si pour un des 4 sens, le compteur vaut au moins 4, le joueur correspondant au dernier jeton a gagné.

3.3. Interface

Il y a 2 versions, une version utilisant une TStringGrid, et une version graphique.

3.3.1. Non graphique

Support

Un TStringGrid est utilisé pour afficher les jetons.

Une ligne supplémentaire de titre est ajoutée, par rapport aux dimensions de la grille du puissance 4, afin de pouvoir afficher le jeton à lacher.

Chaque cellule peut prendre respectivement les valeurs: chaîne vide, lettre 'X', lettre 'O' en fonction de TPlateau.Grille valant respectivement 0, 1, 2.

Affichage d'un jeton

La procédure suivante, montre l'affichage d'un jeton dans le TStringGrid.

Le paramètre ICamp étant le camp du joueur, la valeur 0 efface le jeton.

```
procedure TPlateau.PionAfficher( ICamp: Integer; PPos:
TPoint );
var
  CCar: string;
begin
  case ICamp of
    1 : CCar := ' X';
    2 : CCar := ' O';
    else CCar := '';
  end;
  // le tableau logique commence en indice -1 pour
  le laché de jeton
  StringGrid.Cells[ PPos.X, PPos.Y + 1 ] := CCar;
end;
```

Animation

L'animation est réalisée par la méthode TPlateau.PionGlisser, qui s'occupe d'afficher et d'effacer le jeton successivement de la position initiale à la position finale.

La position finale est récupérée grâce à la méthode TPlateau.CaseVide.

3.4. Conclusion

Un défi amusant et accessible pour beaucoup.

Bravo aux nombreux participants, qui se sont surpassés pour imaginer des améliorations suprenantes :).

Rendez-vous au prochain défi !

TicTacToe le 26/01/2007.

Retrouvez la page de ce défi en entier en ligne, ainsi que la version graphique de TicTacToe : [Lien104](#)

4. La solution du vainqueur

Après la solution du défieur, je vais maintenant vous présenter la solution du vainqueur de ce défi : Cl@udius

4.1. Introduction

Le programme est basé essentiellement sur l'utilisation d'un DrawGrid pour la partie graphique et sur la manipulation de la variable de type TGrille qui représente l'état du 'plateau' sur lequel évoluent les deux joueurs.

4.1.1. DrawGrid

J'ai choisi d'utiliser le composant **DrawGrid** afin de bénéficier le plus simplement possibles de son comportement naturel, notamment les événements SelectCell et DrawCell.

4.1.2. TGrille

Le type TGrille est un tableau d'entiers aux dimensions du jeu Puissance 4 (7 colonnes x 6 lignes).

Ce tableau peut recevoir les valeurs constantes suivantes:

```
const
  VIDE = 0;
  JAUNE = 1;
  ROUGE = -1;
  GAGNANT = 2;
```

4.2. Déroulement du jeu

1. La grille est initialisée avec la valeur VIDE
2. La variable FTour détermine la couleur du joueur qui a la main
3. Un click sur le DrawGrid déclenche l'évènement SelectCell qui dépose un jeton dans la première case vide de la colonne sur laquelle le click a été effectué, puis vérifie si ce jeton est gagnant.
4. Le tour est inversé et retour à l'étape 3, jusqu'à une situation gagnante pour l'un des joueurs ou un match nul.

4.3. Algorithme général

Différentes fonctions contenues dans l'unité uDefs permettent de connaître la situation actuelle sur le plateau.

Notamment :

- **DispoCol** : Nombre de cases vides dans une colonne
- **DispoGrille** : Nombre de cases vides dans la grille
- **PointDeChute** : Calcul du point de chute d'un jeton dans une colonne

Et enfin

- **JoueurGagnant** : Déterminer le dernier jeton déposé donne une situation au joueur

Analyse du dernier jeton joué

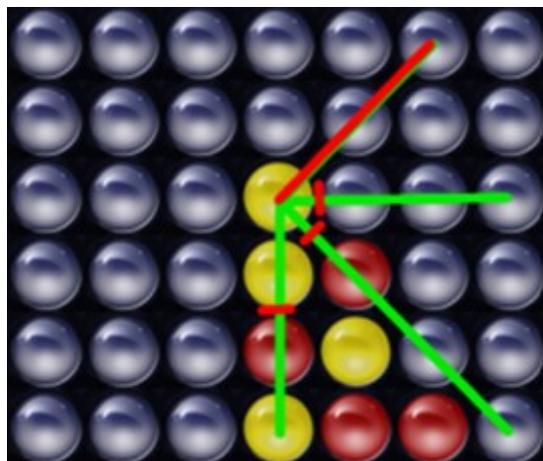
Contrairement à TicTacToe qui analyse uniquement la périphérie de jeton joué, la fonction JoueurGagnant scrute l'ensemble du plateau afin de détecter une situation gagnante pour le joueur.

```
var
  Col, Row: Byte;
begin
  result := False;

  Col := 0;
  while (Col <= MAX_COL) and (not result) do
  begin
    Row := 0;
    while (Row <= MAX_ROW) and (not Result) do
    begin
      result := (DirectionGagnante(1, 0) or //
Horizontal
      DirectionGagnante(0, 1) or // Vertical
      DirectionGagnante(1, 1) or // Diagonal
\
      DirectionGagnante(1,-1)); // Diagonal
/
      Inc(Row);
    end;
    Inc(Col);
  end;
end;
```

La procédure imbriquée 'DirectionGagnante' examine pour chaque direction si le jeton posé est gagnant.

Le comptage s'interrompt dès que la couleur est différente à celle du jeton de référence. Le comptage ne sera pas effectué sur le diagonale / puisque dans ce cas l'alignement de 4 jetons ne peut-être effectué sans sortir de la grille de jeu.



4.4. Partie graphique

Le dessin du plateau est effectué par l'implémentation de l'évènement OnDrawCell du TDrawGrid.

Les images bitmaps : case vide, jeton jaune, rouge ou gagnant (vert) sont inclus au projet comme ressources.

```
procedure TFormMain.GrilleDrawCell(Sender: TObject;
ACol, ARow: Integer;
  Rect: TRect; State: TGridDrawState);
var
  myBitmap: TBitmap;
begin
  myBitmap := TBitmap.Create;
  try
    case FGrille[ACol, ARow] of
      GAGNANT:
myBitmap.LoadFromResourceName(hInstance, 'JETON_VERT');
      JAUNE:
myBitmap.LoadFromResourceName(hInstance,
'JETON_JAUNE');
      ROUGE:
myBitmap.LoadFromResourceName(hInstance,
'JETON_ROUGE');
      else
myBitmap.LoadFromResourceName(hInstance, 'JETON_VIDE');
    end;
    with Grille do
    begin
      Canvas.StretchDraw(Rect, myBitmap);
    end;
  finally
    myBitmap.Free;
  end;
end;
```

4.5. Conclusion

Elaborer un tel programme m'a procuré beaucoup de plaisir, comme je suppose à chacun des participants qui furent particulièrement nombreux sur ce défi.

Rendez-vous au prochain défi !

Cl@udius

Retrouvez la page du vainqueur de ce défi : [Lien105](#)

Les derniers tutoriels et articles

Premiers pas avec Ruby

Ruby est un langage interprété entièrement orienté objet, et multi plate-forme.

Outre le fait d'être très intuitif (vous verrez pourquoi dans la suite de ce cours), Ruby peut servir de langage de script au même titre que PERL ou d'autres, mais son essor récent est surtout dû à l'apparition de Ruby on Rails, un framework web extrêmement puissant basé sur Ruby.

Nous allons voir ici les bases de ce langage, ses atouts, etc.

1. Installation et concepts de base

Voyons comment installer Ruby sous Windows.

Rien de très compliqué, il vous suffit de vous rendre sur la page de Ruby ([Lien106](#)), puis de cliquer sur "Téléchargements", et de télécharger le One-Click Installer pour Windows. Il ne vous reste plus qu'à choisir votre répertoire d'installation, et à valider en attendant la fin de la procédure d'installation.

Ruby est livré par défaut avec deux éditeurs : SciTe et FreeRIDE.

Pour lancer FreeRIDE (intégralement écrit en Ruby), rendez-vous dans le répertoire d'installation de Ruby, puis dans le répertoire "freeride", et exécutez "run.bat".

Je préfère SciTe, mais cela ne dépend que de considérations personnelles. Pour le lancer, rendez-vous dans le répertoire d'installation de Ruby, puis dans le répertoire "scite", et exécutez "SciTE.exe".

En Ruby, tout est objet. Absolument tout. Un entier, une chaîne de caractères, et même les classes, aussi étonnant que cela puisse paraître.

2. Premiers pas

Nous allons apprendre par l'expérience. Pour cela, nous allons utiliser irb (interactive Ruby).

En utilisant irb, nous allons pouvoir tester la syntaxe de base de Ruby et en visualiser directement le résultat, sans avoir à créer un fichier contenant notre code (un .rb) et à le modifier à chaque fois. Il vous suffit de lancer une fenêtre DOS (démarrer -> exécuter -> cmd), puis de taper "irb" dans l'invite de commande, et de valider.

2.1. Manipulation de variables simples

L'invite de commande s'est modifiée. Vous pouvez désormais taper directement vos commandes. Essayez de taper les lignes suivantes :

```
a = 2
b = 3
puts "a + b vaut " + (a+b).to_s
Voici ce que vous devriez obtenir à l'écran :
```

```
irb(main):001:0> a = 2
=> 2
irb(main):002:0> b = 3
=> 3
```

```
irb(main):003:0> puts "a + b vaut " + (a+b).to_s
a + b vaut 5
=> nil
```

Beaucoup de choses dans ce simple exemple. Nous venons de créer deux variables, a et b, et de leur affecter des valeurs. Ensuite, nous avons décidé d'afficher leur somme à l'aide de la méthode puts. Nous affichons donc une chaîne (la partie entre guillemets), que nous concaténons à une autre chaîne à l'aide de l'opérateur +.

Cette deuxième chaîne est en fait le résultat de la somme de nos deux variables (encore cette fois à l'aide de l'opérateur +, qui n'a donc pas la même fonction suivant le type avec lequel il est utilisé), que nous convertissons en chaîne à l'aide de la méthode to_s (pour rendre possible la concaténation, il nous faut deux types identiques).

Vous remarquerez que nulle part nous n'avons défini le type de a ou de b. Ruby se débrouille comme un grand : c'est le duck typing (ça ressemble à un canard, ça se comporte comme un canard, donc c'est un canard).

Les opérateurs sur les nombres restent très classiques : + pour l'addition, - pour la soustraction, * pour la multiplication, et / pour la division.

A noter tout de même ** pour la puissance : a**n signifie a à la puissance n. Ici, n peut être un réel, donc pour obtenir racine de 2, il est tout à fait possible de faire 2**0.5.

Reste la fonction modulo : %, qui renvoie le reste de la division entière.

Ici, nous avons écrit "a+b". Or, en Ruby, tout étant objet, les opérateurs ne sont que des méthodes.

Nous aurions donc pu écrire "a.+(b)" : l'appel à la méthode + de l'objet a avec en paramètre l'objet b.

Peu pratique à écrire, vous en conviendrez, mieux vaut utiliser la première notation. Ruby est bourré de ces sucres syntaxiques qui nous facilitent la vie.

Dans la même veine que to_s, voici quelques fonctions utiles de manipulation de chaîne :

- to_i : convertit une chaîne en un entier, ou retourne la partie entière d'un réel
- to_f : idem, mais retourne un réel
- length : retourne la longueur d'une chaîne ou la longueur d'un tableau
- upcase : convertit une chaîne en majuscules
- downcase : convertit une chaîne en minuscules

- `swapcase` : intervertit les majuscules et les minuscules dans une chaîne
- `capitalize` : met en majuscule le premier caractère de la chaîne
- `reverse` : inverse la chaîne ou les éléments d'un tableau

Jouons encore un peu, et découvrons les chaînes.

Une chaîne peut être définie soit entre apostrophes, soit entre guillemets. Une chaîne définie entre apostrophes ne prendra pas en compte les caractères spéciaux (n par exemple), et n'évaluera pas non plus les expressions entourées par `#{...}` (que nous verrons un peu plus tard, `irb` ne le comprend pas).

Créons donc une chaîne simple :

- `c = 'cou'`
- `puts c * 2`

Et observons la sortie :

```
irb(main):004:0> c = 'cou'
=> "cou"
irb(main):005:0> puts c * 2
=> "coucou"
```

Merveilleux, la multiplication répète notre chaîne !

Voyons maintenant comment demander à l'utilisateur une saisie clavier :

- `saisie = gets`
- `puts saisie.chomp`
- `saisie2 = gets.chomp`

```
irb(main):006:0> saisie = gets
coucou
=> "coucou\n"
irb(main):007:0> saisie.chomp
=> "coucou"
irb(main):008:0> saisie2 = gets.chomp
tralala
=> "tralala"
```

Ici, `gets` permet de demander à l'utilisateur de saisir une valeur. En observant bien, on se rend compte que la frappe sur la touche Entrée est incluse dans la chaîne (le `\n`). Pour l'enlever, rien de plus simple, il suffit d'utiliser la méthode `chomp`, qui renvoie notre chaîne sans le caractère de retour à la ligne.

Lorsque vous souhaitez effectuer le remplacement de variables dans une chaîne en Ruby, il faudra utiliser la syntaxe suivante (notez bien la présence de guillemets pour la chaîne où se produit la substitution) :

```
nom = 'Pierre-Baptiste'
puts "Bonjour #{nom}"
```

Et voici la sortie :

```
Bonjour Pierre-Baptiste
```

Rappel : `irb` ne le comprend pas.

2.2. Tableaux

Nous allons maintenant créer un tableau :

- `mon_tableau = ["a", 12, "b"]`

```
irb(main):001:0> mon_tableau = ["a", 12, "b"]
=> ["a", 12, "b"]
```

Pour accéder aux différents éléments de notre tableau, il suffit de faire suivre le nom de notre variable de crochets, avec l'indice désiré entre les crochets. Le premier indice du tableau est 0 :

- `puts mon_tableau[1]`
- `mon_tableau[1] = 42`
- `puts mon_tableau[1]`

```
irb(main):002:0> puts mon_tableau[1]
12
=> nil
irb(main):003:0> mon_tableau[1] = 42
=> 42
irb(main):004:0> puts mon_tableau[1]
42
=> nil
```

Il existe de nombreuses autres méthodes qui s'appliquent aux tableaux : `push`, `pop`, `last`, etc.

```
irb(main):005:0> tablo = ['a', 'z', 'e', 'r', 't']
=> ["a", "z", "e", "r", "t"]
irb(main):006:0> tablo.push('y', 'u')
=> ["a", "z", "e", "r", "t", "y", "u"]
irb(main):007:0> tablo.pop
=> "u"
irb(main):008:0> tablo
=> ["a", "z", "e", "r", "t", "y"]
irb(main):009:0> tablo.last
=> "y"
irb(main):010:0> tablo
=> ["a", "z", "e", "r", "t", "y"]
```

Juste une petite méthode fort sympathique pour en finir avec les tableaux : la méthode `sort`. Faites juste un `mon_tableau.sort`, et vous verrez alors votre tableau trié :

2.3. Hash / tableaux associatifs

Ruby gère également les tableaux associatifs. Il faut juste noter qu'en Ruby, ils sont appelés "hash".

- `mon_hash = {:nom => "Naigeon", :prenom => "PB"}`

```
irb(main):001:0> mon_hash = {:nom => "Naigeon", :prenom
=> "PB"}
=> {:nom => "Naigeon", :prenom => "PB"}
```

Accéder aux éléments n'est guère plus compliqué :

```
puts mon_hash[:nom]
puts mon_hash[:prenom]
mon_hash[:prenom] = "Pierre-Baptiste"
puts mon_hash[:prenom]
```

```
irb(main):002:0> puts mon_hash[:nom]
Naigeon
=> nil
irb(main):003:0> puts mon_hash[:prenom]
PB
=> nil
irb(main):004:0> mon_hash[:prenom] = "Pierre-Baptiste"
=> "Pierre-Baptiste"
irb(main):005:0> puts mon_hash[:prenom]
Pierre-Baptiste
=> nil
```

2.4. Manipulation de chaînes, de tableaux et de hash

Il est possible d'accéder aux différentes parties de la chaîne comme suit :

```
irb(main):001:0> ma_var = 'azerty'  
=> "azerty"  
irb(main):002:0> ma_var[0]  
=> 97  
irb(main):003:0> ma_var[0..0]  
=> "a"  
irb(main):004:0> ma_var[0..3]  
=> "azer"  
irb(main):005:0> ma_var[0,4]  
=> "azer"  
irb(main):006:0> ma_var[-2,2]  
=> "ty"
```

Dans le premier cas, nous obtenons le code ASCII de la première lettre (l'indice 0).

Ensuite, nous lui demandons d'afficher les lettres comprises entre l'indice 0 et l'indice 0 (les bornes sont incluses). Il n'y en a qu'une, il nous renvoie donc "a".

Puis nous lui demandons donc d'afficher les lettres comprises entre l'indice 0 et l'indice 3 (quatre lettres donc). Il nous retourne "azer".

Nous lui demandons ensuite de nous afficher à partir de l'indice 0 une chaîne de longueur 4. Il nous retourne donc encore "azer".

Pour finir, nous lui passons un indice négatif. Les indices négatifs partent de la fin du tableau (et commencent à 1, eux, puisqu'il est dur de différencier 0 et -0). Nous lui demandons donc d'afficher à partir du deuxième caractère avant la fin une chaîne de longueur 2 (les deux derniers caractères quoi !).

Il est possible de procéder de même avec les tableaux :

```
irb(main):007:0> mon_tabl = ['q', 'w', 'e', 'r', 't', 'y']  
=> ["q", "w", "e", "r", "t", "y"]  
irb(main):008:0> mon_tabl[0]  
=> "q"  
irb(main):009:0> mon_tabl[0..0]  
=> ["q"]  
irb(main):010:0> mon_tabl[0..3]  
=> ["q", "w", "e", "r"]  
irb(main):011:0> mon_tabl[0,4]  
=> ["q", "w", "e", "r"]  
irb(main):012:0> mon_tabl[-2,2]  
=> ["t", "y"]
```

Cela fonctionne exactement comme précédemment. Faites juste attention au fait que `mon_tabl[0]` retourne une chaîne (dans le cas présent), alors que `mon_tabl[0..0]` retourne cette même chaîne, mais dans un tableau !

Encore deux petites fonctions extrêmement utiles avant de conclure cette section : `split` et `join` (équivalents de `explode` et `implode` en PHP).

```
irb(main):010:0> ma_date = '18/02/1981'  
=> "18/02/1981"  
irb(main):011:0> mon_tableau = ma_date.split('/')  
=> ["18", "02", "1981"]  
irb(main):012:0> ma_date2 = mon_tableau.join('-')  
=> "18-02-1981"
```

Vous pouvez bien entendu spécifier le séparateur de votre choix. `Split` vous permet donc de découper une chaîne dans un tableau en

fonction d'un séparateur donné, et `join` vous permet de regrouper les éléments d'un tableau dans une chaîne, toujours avec un séparateur.

A vous maintenant la manipulation de chaînes :)

2.5. Conventions de nommage

Variables et méthodes

- `nom_var`
- `nom_methode`
- `nom_methode?` (une méthode qui renvoie true ou false : `is_a?(Fixnum)`)
- `nom_methode!` (une méthode "destructrice", souvent qui modifie la variable : `sort!`)
- `@nom_var_instance`
- `@@nom_var_classe`
- `$nom_var_globale` (Attention, peut aussi s'écrire en majuscules : `$NOM_VARIABLE_GLOBALE`)

Classes

- `NomClasse`

Constantes

- `NOMCONSTANTE` ou `NomConstante`

2.6. Expressions régulières

Ruby est tout à fait de taille à jouer avec les expressions régulières. En revanche, moi, je suis loin d'être de taille à jouer avec elles, c'est pourquoi vous ne disposerez ici que d'un petit exemple, à vous de faire la suite :)

```
chaine = 'la couleur #FF024D est-elle une couleur  
HTML ?'
```

```
test = (chaine =~ /[0-9A-Fa-f]{6}/)
```

```
if (test)  
  puts "la chaîne #{chaine[test,7]} est un code couleur  
  HTML valide !"  
else  
  puts "Pas de code couleur HTML valide dans la chaîne"  
end
```

Nous nous contentons simplement de parcourir notre chaîne à la recherche du motif, et nous renvoyons sa position si le motif est trouvé.

Notez simplement l'opérateur "`=~`". C'est l'opérateur réservé à l'évaluation d'expressions régulières. Il retourne la position du premier caractère qui correspond dans la chaîne, nil si rien n'est trouvé.

3. Les structures de contrôle

Pour les différentes structures de contrôle ci-dessous, je vous montrerai simplement un exemple d'utilisation, je ne pense pas qu'une explication soit nécessaire :)

D'une manière générale, les structures de contrôle s'appliquent à tout ce qui est compris entre le nom de l'instruction (`if`, `while`, ...) et le mot-clé `end` (parfois les instructions seront comprises entre accolades `{...}`).

3.1. Opérateurs de comparaison

En Ruby, les opérateurs classiques de comparaison s'appliquent :

- `a == b` : a égal b (attention, contrairement à PHP par exemple, `'42' == 42` renverra false)
- `a != b` : a différent de b

- $a < b$: a strictement inférieur à b
- $a > b$: a strictement supérieur à b
- $a \leq b$: a inférieur ou égal à b
- $a \geq b$: a supérieur ou égal à b

3.2. if ... else ... end

```
puts "Merci de saisir un nombre : "
chiffre = gets.chomp.to_i
if (chiffre == 42)
  puts "Bingo, LA réponse !"
else
  puts "Essaie encore"
end
```

Notez simplement le `to_i`, qui transforme la valeur récupérée est un entier.

Attention : Dans le cas d'une suite de caractères alphanumériques, `to_i` a un comportement spécifique :

```
irb(main):001:0> "42".to_i
=> 42
irb(main):002:0> "42abc".to_i
=> 42
irb(main):003:0> "abc42abc".to_i
=> 0
irb(main):004:0> "abc".to_i
=> 0
```

Tant que les caractères sont des chiffres, Ruby les prend en compte. Dès qu'il rencontre autre chose, il s'arrête. Dans le cas où le premier caractère de la chaîne ne correspond pas, il nous retourne 0.

3.3. case ... end

Le `case` permet d'effectuer différentes actions en fonction de la valeur d'un paramètre sans avoir à imbriquer trop de `if`.

```
puts "Merci de saisir un nombre : "
chiffre = gets.chomp.to_i
case chiffre
  when 0..9
    puts "Vous aimez les tout petits chiffres."
  when 42
    puts "Vous être un as !"
  else
    puts "Vous auriez mieux pu choisir... ;)"
end
```

La seule chose importante à noter ici est l'intervalle spécifié dans le "when 0..9".

3.4. unless ... end

Un `unless` est simplement un `if` à l'envers : sauf si :

```
j = rand(10)
unless (j <= 5)
```

```
puts "j vaut plus de 5"
end
```

3.5. while ... end

```
puts "Je vais compter de 1 a 10"
i = 0
while (i < 10)
  i += 1
  puts i
end
```

A noter la méthode d'incrémement... inutile d'essayer `i++`, qui ne ferait que vous générer une erreur.

3.6. until ... end

Le `until` est juste un `while` qui marche dans l'autre sens. En reprenant l'exemple précédent :

```
puts "Je vais compter de 1 a 10"
i = 0
until (i >= 10)
  puts i
  i += 1
end
```

3.7. for ... end

La boucle `for` est un peu particulière, puisqu'elle s'applique à des éléments d'une collection :

```
puts "compte de 1 a 10"
for elt1 in (1..10)
  puts elt1
end
```

```
tableau = [1, 2, 3, 4]
for elt2 in tableau
  puts elt2 + 1
end
```

3.8. Sortie de boucles

Il existe différents moyens de sortir d'une boucle : "return", "break", "next" et "redo". Voyons les spécificités de chacun :

- `return` : sort complètement de la boucle et de la méthode qui contient la boucle.
- `break` : sort de la boucle, et continue le code juste après.
- `next` : passe directement à la prochaine itération de la boucle
- `redo` : re-passe l'itération courante de la boucle

Attention : Il est facile à l'aide de ces commandes de réaliser des boucles infinies. Prenez donc le temps de bien regarder comment elles fonctionnent.

Retrouvez l'article complet de Pierre-Baptiste Naigeon en ligne et apprenez tout ce qu'il vous faut pour démarrer avec Ruby : [Lien107](#)

Les livres Ruby

Pratique de Ruby on Rails

Le succès foudroyant de Ruby on Rails rappelle que les

développeurs Web exigent dorénavant de leurs outils qu'ils s'installent aisément sur toutes les plates-formes, qu'ils minimisent le nombre de lignes de code à écrire (et à maintenir), et qu'ils

offrent un environnement de test et de mise au point irréprochable.

Pour tirer le meilleur parti de Rails, il est indispensable de se familiariser avec Ruby, de connaître les règles qui président à la conception d'une base de données relationnelle, et de faire sien le fameux modèle MVC (Modèles, Vues, Contrôleurs). Éric Sarrion nous explique tout cela avec sa minutie habituelle, dans un ouvrage très progressif.

Pratique de Ruby on Rails se compose de 5 parties :

- La partie Introduction montre en 15 pages comment installer Rails et créer une petite application. Elle plonge ensuite dans les bases de la programmation Ruby, un langage qui s'avère particulièrement agréable à apprendre.
- La partie Contrôleurs montre comment écrire les actions destinées à être appelées par le serveur http. C'est là que le développeur décide de la réaction de l'application en fonction de l'url demandée.
- Dans la partie Modèles, vous apprendrez non seulement à créer et modifier les objets qui représentent votre base de données, mais aussi à concevoir une base de données relationnelle dans les règles de l'art.
- La partie Vues se concentre sur le rendu des valeurs calculées par les contrôleurs et les objets du modèle. C'est là que vous apprendrez à paginer votre affichage HTML, intégrer CSS et JavaScript, et bien sûr utiliser Ajax.
- La dernière partie aborde des fonctions indispensables comme l'envoi de mail ou la publication de services Web, sans lesquels une application web moderne ne serait pas complète.

Voici donc un manuel d'apprentissage progressif et très complet, destiné à tous ceux, du programmeur occasionnel au développeur professionnel, qui souhaitent se mettre rapidement à Rails sans en maîtriser nécessairement tous les pré-requis.

Critique du livre par la rédaction (Gérard Ernaelsten)

Le livre est considéré pour un public débutant à avancé. Pour ma part je viens du monde « Php », et je pense que quelques notions en « programmation orientée objet » sont nécessaires pour aborder ce livre.

L'auteur Eric Sarrion nous emmène dans les méandres de « Ror » avec une facilité déconcertante, ce livre est composé de cinq parties :

- introduction
- les contrôleurs
- les modèles
- les vues
- compléments
- Introduction

Dans cette première partie, Eric Sarrion nous fait un rappel de l'installation de "Ruby on Rails" sur différentes architectures.

Il nous propose également une première application "Rails", de façon à nous mettre directement dans le bain. Cela est fait de façon claire et concise en abordant quelques points essentiels comme le "Scaffold" et "WEBrick".

Les soixante-dix pages suivantes sont consacrées au langage "Ruby" lui même. On passe en revue l'essentiel de cette programmation de la création de variable à la gestion des exceptions, de la table de hachage (très bien expliquée et importante pour comprendre les "::params"), aux boucles et

itérateurs. La partie sur les modules pourrait être, je pense, plus claire.

Arrivé à la fin de ce chapitre vous êtes fichu, il faut aller jusqu'au bout :))

Les contrôleurs

Cette partie du livre pose les bases du succès de "RoR".

On nous explique les relations qu'il y a entre les "Url" et les définitions dans les classes contrôleurs.

Le premier chapitre de cette partie du livre est consacré à la définition des "Url" définies dans le fichier de routage(config/routes.rb).

Même si au départ cela ne semble pas très clair, l'auteur n'a de cesse durant tout l'ouvrage de nous faire référence (en indiquant chaque fois la page) à ses propos, ce qui donne un réel fil conducteur durant toute la lecture.

Le nombre de choses qu'il passe en revue est assez conséquent, en vrac nous passons de la méthode "render()", aux filtres, en ayant vu au passage les sessions et les flash.

Nous pourrions regretter que l'auteur, E.Sarrion, se contente de ne rester que dans un seul contrôleur et de ne faire de l'interaction que dans cet unique contrôleur (même si on comprend bien la logique), cela m'a posé problème notamment lors des essais avec les sessions lorsque l'exemple de l'auteur ne fonctionne correctement que si l'on reste dans le même contrôleur.

Les modèles

Le chapitre sur les modèles est impressionnant, il représente à lui seul à peu près un tiers du livre.

On passe en revue tout ce qui concerne l'accès aux bases de données.

Les systèmes de connection (par défaut, ou dans un modèle), la création des structures de tables, la recherche dans les tables, ainsi que les modifications etc.

On a même droit à un petit cours sur les entités/associations et la façon de modéliser nos tables.

Tout est très bien expliqué, c'est clair, c'est concis, c'est agréable à lire, les méthodes "Rails" sont vues dans leur ensemble, ainsi que les possibilités de code "SQL" avec la méthode "find_by_sql()" par exemple.

Nous savons comment gérer les contraintes de cardinalité grâce à Rails, nous apprenons les mécanismes de validation des données.

Le cycle de vie d'un objet et les transactions sont également abordés dans cette partie du livre.

Les vues

Dans cette rubrique on nous explique la coopération entre "contrôleurs et vues" et entre "modèles et vues".

Dans les différents chapitres nous apprenons comment visualiser nos actions, comment partager notre codes dans l'architecture "MVC".

L'auteur, dans sa pratique pédagogique ne se contente pas de nous donner des recettes miracles, mais quand il propose un exemple, il le mettra toujours sous forme "HTML" et sous forme "Rails", en

nous expliquant bien les avantages de l'un et de l'autre.

De même si vous devez créer une table, le code Sql et le code Rails seront chaque fois exposés de sorte que vous puissiez choisir une méthode ou l'autre.

Comme je vous l'ai signalé, l'auteur se contente de rester dans un programme minimaliste, toutefois dans cette partie du livre, il développe deux petites applications (mise à jour d'une base client et la gestion d'une agence immobilière) qui vont nous permettre d'apprécier toute l'étendue et la puissance de "Ror".

Un chapitre est consacré au "CSS", mais ce qui est plus intéressant encore, un chapitre qui est dédié au partage de code entre les vues, les contrôleurs et les modèles, grâce aux fichiers assistants (helper).

Il y a également un chapitre intéressant et bien fourni sur "Ajax" et les composants de script.aculo.us.

Compléments

Dans cette partie du livre l'auteur nous parle de la gestion de

mails, et des services Web.

Il y également une brève introduction au serveur web "WEBrick", ainsi qu'à l'IDE RadRails (E.Sarrion préfère Scite).

Nous finirons cet ouvrage par quelques conseils sur la méthodologie de conception d'un projet Rails.

Conclusion

Ma conclusion est que ce livre est excellent, pour entrer dans le monde "Ror". Notez que c'est le premier livre que je lis sur ce langage!

Néanmoins ayant un projet à développer, j'ai bien l'ensemble des bases pour débiter et approfondir maintenant les connaissances acquises tout au long de cette lecture.

Sans hésiter je recommande ce livre pour celui ou celle qui veut en savoir plus sur "Ruby on Rails".

Retrouvez ce livre sur la rubrique Ruby : [Lien108](#)

FAQ Ruby

Comment personnaliser ma page d'accueil ?

Supposons que vous vouliez afficher directement votre controller admin(app/controller/admin).

Supprimez le fichier "index.html" dans le dossier "public".

Modifier le fichier "/config/routes.rb" et commentez la ligne suivante (si cela n'est déjà pas fait) :

```
map.connect '', :controller => "welcome"
```

Ajoutez la ligne suivante (toujours dans le fichier "/config/routes.rb") :

```
map.connect '', :controller => "admin"
```

Au prochain démarrage à l'adresse http://localhost:3000, votre controller admin apparaîtra.

Comment créer ses tables en utilisant les fichiers de migration ?

Pour créer une table "utilisateurs", placez-vous dans le répertoire de votre application Rails, et tapez la ligne de commande suivante :

```
ruby script/generate model utilisateur
```

Rails vous générera alors plusieurs fichiers :

```
create app/models/utilisateur.rb
create test/unit/utilisateur_test.rb
create test/fixtures/utilisateurs.yml
exists db/migrate
create db/migrate/XXX_create_utilisateurs.rb
```

Avec XXX un numéro de migration autoincrémenté.

Il ne vous reste plus qu'à effectuer votre migration.

Comment migrer d'une version de base de données à une autre ?

Dans le répertoire de votre application Rails, tapez la ligne suivante en ligne de commande :

```
rake db:migrate VERSION=XXX
```

Avec XXX le numéro de version vers lequel vous souhaitez migrer (dans un sens ou dans l'autre).

A noter tout de même : Rails exécutera tous les fichiers de migration situés entre le numéro de migration actuel et le numéro de migration final.

Retrouvez la FAQ Ruby en ligne : [Lien109](#)

Les derniers tutoriels et articles

Implémentation du pattern MVC en Java

MVC est un design pattern très puissant, mais néanmoins assez complexe, qui permet de bien structurer de grosses applications graphiques. Nous allons apprendre à la maîtriser avec ce tutorial.

1. Le design pattern MVC

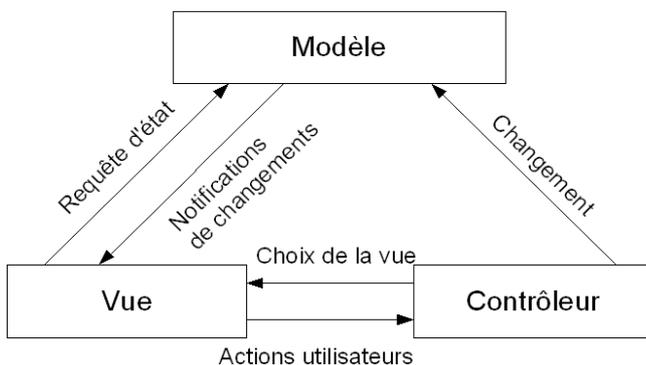
Le design pattern Modèle-Vue-Contrôleur (MVC) est un pattern architectural qui sépare les données (le modèle), l'interface homme-machine (la vue) et la logique de contrôle (le contrôleur).

Ce modèle de conception impose donc une séparation en 3 couches :

- Le modèle : Il représente les données de l'application. Il définit aussi l'interaction avec la base de données et le traitement de ces données.
- La vue : Elle représente l'interface utilisateur, ce avec quoi il interagit. Elle n'effectue aucun traitement, elle se contente simplement d'afficher les données que lui fournit le modèle. Il peut tout à fait y avoir plusieurs vues qui présentent les données d'un même modèle.
- Le contrôleur : Il gère l'interface entre le modèle et le client. Il va interpréter la requête de ce dernier pour lui envoyer la vue correspondante. Il effectue la synchronisation entre le modèle et les vues.

La synchronisation entre la vue et le modèle se passe avec le pattern Observer. Il permet de générer des événements lors d'une modification du modèle et d'indiquer à la vue qu'il faut se mettre à jour.

Voici un schéma des interactions entre les différentes couches :



Ce modèle de conception permet principalement 2 choses :

- Le changement d'une couche sans altérer les autres. C'est-à-dire que comme toutes les couches sont clairement séparées, on doit pouvoir en changer une pour, par exemple, remplacer Swing par SWT sans porter atteinte aux autres couches. On pourrait aussi donc changer le modèle sans toucher à la vue et au contrôleur. Cela rend les modifications plus simples.
- La synchronisation des vues. Avec ce design pattern,

toutes les vues qui montrent la même chose sont synchronisées.

Il faut tout de même garder en mémoire, que la mise en oeuvre de MVC dans une application n'est pas des plus simples. En effet, ce modèle de conception introduit tout de même un niveau de complexité assez élevé. De plus, implémenter MVC dans votre application nécessite une bonne conception dès le départ. Ce qui peut prendre du temps. Ce pattern n'est donc à conseiller que pour les moyennes et grandes applications.

2. Implémentation

Nous allons choisir un exemple très simple (voire même simpliste). Ce sera tout simplement une application permettant de modifier un volume. Il y aura plusieurs vues pour représenter ce volume et après toute modification, les vues devront être synchronisées. L'interface sera développée avec Swing.

C'est certes bidon et très petit comme exemple, mais cela permettra de voir simplement la séparation des couches et l'utilisation du design pattern Observer.

Par souci de simplicité, nous n'utiliserons ici qu'un seul package appelé volume. Dans une grosse application MVC, je vous conseille plutôt de clairement séparer le modèle, les vues et l'interface dans des packages différents.

2.1. Le modèle

Le modèle est assez simple à développer, vu qu'il ne gère qu'un volume. Pour commencer, on va déjà développer la base de notre modèle :

VolumeModel.java

```
public class VolumeModel {
    private int volume;

    public VolumeModel() {
        super();

        volume = 0;
    }

    public int getVolume() {
        return volume;
    }

    public void setVolume(int volume) {
        this.volume = volume;
    }
}
```

Voilà, la première fonction de notre modèle est remplie, il fournit maintenant un volume qui peut être modifié.

Mais maintenant, il faut que notre modèle puisse notifier un changement de volume. Pour cela, on va employer les listeners. On va donc créer un nouveau listener (`VolumeListener`) et un nouvel événement (`VolumeChangedEvent`) :

VolumeListener.java

```
import java.util.EventListener;

public interface VolumeListener extends EventListener {
    public void volumeChanged(VolumeChangedEvent
event);
}
```

VolumeChangedEvent.java

```
import java.util.EventObject;

public class VolumeChangedEvent extends EventObject{
    private int newVolume;

    public VolumeChangedEvent(Object source, int
newVolume){
        super(source);

        this.newVolume = newVolume;
    }

    public int getNewVolume(){
        return newVolume;
    }
}
```

Maintenant, nous allons implémenter ce système d'écouteurs dans le modèle pour que d'autres entités puissent "écouter" les changements du modèle.

VolumeModel.java

```
import javax.swing.event.EventListenerList;

public class VolumeModel {
    private int volume;

    private EventListenerList listeners;

    public VolumeModel(){
        this(0);
    }

    public VolumeModel(int volume){
        super();

        this.volume = volume;

        listeners = new EventListenerList();
    }

    public int getVolume() {
        return volume;
    }

    public void setVolume(int volume) {
        this.volume = volume;

        fireVolumeChanged();
    }

    public void addVolumeListener(VolumeListener
listener){
```

```
listeners.add(VolumeListener.class,
listener);
    }

    public void removeVolumeListener(VolumeListener
l){
        listeners.remove(VolumeListener.class,
l);
    }

    public void fireVolumeChanged(){
        VolumeListener[] listenerList =
(VolumeListener[])listeners.getListeners(VolumeListener
.class);

        for(VolumeListener listener :
listenerList){
            listener.volumeChanged(new
VolumeChangedEvent(this, getVolume()));
        }
    }
}
```

Voilà, maintenant notre modèle avertit tous ses écouteurs à chaque changement de volume. Ensuite, en fonction de l'application, on peut tout à fait imaginer plusieurs listeners par modèles et d'autres événements dans les listeners, par exemple quand le volume dépasse certains seuils... Vous voyez donc qu'un modèle peut très vite devenir conséquent.

2.2. Le contrôleur

Nous allons maintenant développer notre contrôleur. Comme le contrôleur doit le moins possible être dépendant de Swing, on va créer une classe abstraite représentant une vue du volume.

VolumeView.java

```
public abstract class VolumeView implements
VolumeListener{
    private VolumeController controller = null;

    public VolumeView(VolumeController controller){
        super();

        this.controller = controller;
    }

    public final VolumeController getController(){
        return controller;
    }

    public abstract void display();
    public abstract void close();
}
```

Maintenant notre contrôleur ne manipulera que des objets de type View et non plus de type Swing.

A nouveau dans un souci de simplicité, nous allons créer un seul contrôleur pour les 3 vues que nous aurons. Dans notre cas, c'est plus simple de faire ainsi, vu que nos 3 vues font toutes la même chose et font très peu de choses. Dans le cas de vue fondamentalement différentes, il est fortement conseillé d'utiliser plusieurs contrôleurs.

VolumeController.java

```
public class VolumeController {
    public VolumeView fieldView = null;
    public VolumeView spinnerView = null;
    public VolumeView listView = null;
}
```

```

private VolumeModel model = null;

public VolumeController (VolumeModel model){
    this.model = model;

    fieldView = new JFrameFieldVolume(this,
model.getVolume());
    spinnerView = new
JFrameSpinnerVolume(this, model.getVolume());
    listView = new JFrameListVolume(this,
model.getVolume());

    addListenersToModel();
}

private void addListenersToModel() {
    model.addVolumeListener(fieldView);
    model.addVolumeListener(spinnerView);
    model.addVolumeListener(listView);
}

public void displayViews(){
    fieldView.display();
    spinnerView.display();
    listView.display();
}

public void closeViews(){
    fieldView.close();
    spinnerView.close();
    listView.close();
}

public void notifyVolumeChanged(int volume){
    model.setVolume(volume);
}
}

```

2.3. Les vues

Comme nous l'avons vu avec le développement du contrôleur, nous allons avoir 3 vues :

- Une vue permettant de modifier le volume avec un champ texte avec un bouton permettant de valider le nouveau volume : JFrameFieldVolume
- Une vue permettant de modifier le volume à l'aide d'un spinner avec un bouton permettant de valider le nouveau volume : JFrameSpinnerVolume
- Une vue listant les différents volumes et qui ajoutera chaque nouveau volume dans une liste déroulante : JFrameListVolume

Toutes ces vues seront représentées par une JFrame.

Voici donc nos 3 vues :

JFrameFieldVolume.java

```

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.text.NumberFormat;

import javax.swing.JButton;
import javax.swing.JFormattedTextField;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.text.DefaultFormatter;

public class JFrameFieldVolume extends VolumeView
implements ActionListener{
    private JFrame frame = null;

```

```

private JPanel contentPane = null;
private JFormattedTextField field = null;
private JButton button = null;
private NumberFormat format = null;

public JFrameFieldVolume (VolumeController
controller) {
    this(controller, 0);
}

public JFrameFieldVolume (VolumeController
controller, int volume){
    super(controller);

    buildFrame(volume);
}

private void buildFrame(int volume) {
    frame = new JFrame();

    contentPane = new JPanel();

    format =
NumberFormat.getNumberInstance();
    format.setParseIntegerOnly(true);
    format.setGroupingUsed(false);
    format.setMaximumFractionDigits(0);
    format.setMaximumIntegerDigits(3);

    field = new JFormattedTextField(format);
    field.setValue(volume);

    ((DefaultFormatter) field.getFormatter()).setAllowsInvalid(false);

    contentPane.add(field);

    button = new JButton("Mettre à jour");
    button.addActionListener(this);
    contentPane.add(button);

    frame.setContentPane(contentPane);
    frame.setTitle("JFrameSpinnerVolume");
    frame.pack();
}

public void close() {
    frame.dispose();
}

public void display() {
    frame.setVisible(true);
}

public void volumeChanged (VolumeChangedEvent
event) {
    field.setValue(event.getNewVolume());
}

public void actionPerformed (ActionEvent arg0) {
    getController().notifyVolumeChanged(Integer.parseInt(field.getValue().toString()));
}
}

```

JFrameListVolume.java

```

import javax.swing.DefaultListModel;
import javax.swing.JFrame;
import javax.swing.JList;
import javax.swing.JPanel;

```

```

import javax.swing.JScrollPane;

public class JFrameListVolume extends VolumeView {
    private JFrame frame = null;
    private JPanel contentPane = null;
    private JList listVolume = null;
    private JScrollPane scrollVolume = null;
    private DefaultListModel jListModel = null;

    public JFrameListVolume (VolumeController
controller) {
        this(controller, 0);
    }

    public JFrameListVolume (VolumeController
controller, int volume){
        super(controller);

        buildFrame (volume);
    }

    private void buildFrame(int volume) {
        frame = new JFrame();

        contentPane = new JPanel();

        jListModel = new DefaultListModel();
        jListModel.addElement (volume);

        listVolume = new JList(jListModel);

        scrollVolume = new
JScrollPane(listVolume);
        contentPane.add(scrollVolume);

        frame.setContentPane(contentPane);
        frame.setTitle("JFrameListVolume");
        frame.pack();
    }

    public void close() {
        frame.dispose();
    }

    public void display() {
        frame.setVisible(true);
    }

    public void volumeChanged (VolumeChangedEvent
event) {

        jListModel.addElement (event.getNewVolume());
    }
}

```

JFrameSpinnerVolume.java

```

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JSpinner;
import javax.swing.SpinnerNumberModel;

public class JFrameSpinnerVolume extends VolumeView
implements ActionListener{
    private JFrame frame = null;
    private JPanel contentPane = null;
    private JSpinner spinner = null;
    private SpinnerNumberModel spinnerModel = null;

```

```

private JButton button = null;

    public JFrameSpinnerVolume (VolumeController
controller) {
        this(controller, 0);
    }

    public JFrameSpinnerVolume (VolumeController
controller, int volume){
        super(controller);

        buildFrame (volume);
    }

    private void buildFrame(int volume) {
        frame = new JFrame();

        contentPane = new JPanel();

        spinnerModel = new
SpinnerNumberModel (volume, 0, 100, 1);

        spinner = new JSpinner(spinnerModel);
        contentPane.add(spinner);

        button = new JButton("Mettre à jour");
        button.addActionListener (this);
        contentPane.add(button);

        frame.setContentPane(contentPane);
        frame.setTitle("JFrameSpinnerVolume");
        frame.pack();
    }

    public void close() {
        frame.dispose();
    }

    public void display() {
        frame.setVisible(true);
    }

    public void volumeChanged (VolumeChangedEvent
event) {

        spinnerModel.setValue (event.getNewVolume());
    }

    public void actionPerformed (ActionEvent arg0) {

        getController().notifyVolumeChanged (spinnerModel.getNumber().intValue());
    }
}

```

Voilà, nos 3 vues sont maintenant développées. Elles sont tout ce qu'il y a de plus basiques et on pourrait bien sûr les optimiser, en les plaçant correctement sur l'écran, en donnant une bonne taille aux composants, ... Mais là n'est pas le but de ce tutorial.

2.4. La classe lanceur

On va maintenant créer la classe "main" de l'application. Sa fonction est plus que simple, elle crée un nouveau modèle, crée un nouveau contrôleur en lui passant le modèle et demande au contrôleur d'afficher les vues.

Voici donc à quoi elle va ressembler :

```

JVOLUME.java
public class JVOLUME {

```

```

public static void main(String[] args) {
    VolumeModel model = new VolumeModel(50);
    VolumeController controller = new
VolumeController(model);
    controller.displayViews();
}
}

```

Rien de bien compliqué donc.

2.5. Résultat

Voilà, notre exemple est maintenant terminé. Vous voyez que nous avons pu rapidement mettre en oeuvre une architecture MVC, mais que ca augmente tout de même le temps de développement et la taille du code et que nous sommes dans une toute petite application. Mais le résultat est tout de même là, toutes nos vues sont synchronisées et le moindre changement sur une des vues est tout de suite actifs sur les autres vues.

Avec cet architecture, si nous voulons ajouter une nouvelle vue, il suffit tout simplement d'ajouter 4 lignes dans le contrôleur (une pour la déclaration, une pour l'initialisation et une autre pour les méthodes close et display).

3. Changement de framework graphique

Nous allons changer de framework graphique pour une des vues. Nous allons donc modifier notre JFrameListVolume pour la faire passer en AWT. C'est donc une nouvelle classe FrameListVolume qui va venir la remplacer.

Le premier changement à faire est de remplacer dans le constructeur du contrôleur JFrameListVolume par FrameListVolume :

Constructeur du contrôleur

```

public VolumeController (VolumeModel model){
    this.model = model;

    fieldView = new JFrameFieldVolume(this,
model.getVolume());
    spinnerView = new JFrameSpinnerVolume (this,
model.getVolume());
    listView = new FrameListVolume (this,
model.getVolume());

    addListenersToModel ();
}

```

Ensuite nous allons développer notre interface avec AWT :

FrameListVolume.java

```

import java.awt.Frame;
import java.awt.List;

public class FrameListVolume extends VolumeView {
    private Frame frame = null;
    private List listVolume = null;

    public FrameListVolume (VolumeController
controller) {
        this(controller, 0);
    }

    public FrameListVolume (VolumeController
controller, int volume){
        super (controller);

```

```

        buildFrame (volume);
    }

    private void buildFrame (int volume) {
        frame = new Frame ();

        listVolume = new List ();

        listVolume.add (Integer.toString (volume));
        frame.add (listVolume);

        frame.setTitle ("JFrameListVolume");
        frame.pack ();
    }

    public void close () {
        frame.dispose ();
    }

    public void display () {
        frame.setVisible (true);
    }

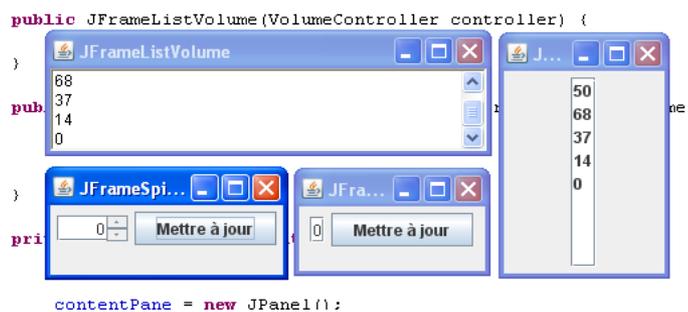
    public void volumeChanged (VolumeChangedEvent
event) {
        listVolume.add (Integer.toString (event.getNewVolume ()));
    }
}

```

Eh voilà, nous avons migré une partie de notre application vers AWT sans le moindre problème. Et grâce à notre architecture, nous n'avons dû changer qu'une petite partie du contrôleur et la partie présentation.

Maintenant, pour ne pas avoir développé la première vue Swing avec la liste pour rien, nous allons l'ajouter à nouveau au contrôleur. Nous avons donc maintenant quatre vues dans notre application et nous avons ajouté la vue avec une simplicité déroutante.

Voilà ce que donnera donc notre application au final :



4. Conclusion

Voilà, vous venez de développer votre première (ou une nouvelle pour certains) application respectant l'architecture MVC. Comme vous avez pu le voir, cette architecture permet une très bonne séparation des couches et facilite l'ajout ou la modification de vues. Par contre, elle nécessite un travail supplémentaire et augmente la quantité de code à écrire. Elle n'est donc destinée qu'à de grosses applications ou à de moyennes applications en évolution constante. Mais rien ne vous empêche de l'utiliser dans n'importe quel développement.

Retrouvez l'article de Baptiste Wicht en ligne : [Lien110](#)

Les derniers tutoriels et articles

Présentation de la bibliothèque IrrKlang

Cet article va présenter les fonctionnalités de la bibliothèque de gestion audio irrKlang, j'y présenterais ses atouts et ses points faibles.

1. Installation

A l'heure où j'écris cet article, irrKlang en est encore à la version 0.5, tout n'est pas finalisé mais la structure générale de la bibliothèque est déjà bien visible.

IrrKlang est une bibliothèque C++ de gestion audio, elle permet de jouer des sons aux formats mp3, ogg et wav dans un espace 3D. IrrKlang est multi-plateforme, elle utilise DirectSound sous Windows et ALSA sous Linux.

Pour installer IrrKlang, rendez-vous sur le site officiel ([Lien111](#)) puis téléchargez le pack. Le même pack servira sous Windows ou sous Linux. Le pack contient la bibliothèque compilée en statique, la bibliothèque dynamique, les headers nécessaires à la compilation ainsi que des exemples de code.

Pour compiler, il suffira d'ajouter

```
#include <irrklang.h>
```

Ainsi, le répertoire include du pack irrklang doit être accessible depuis votre compilateur. Et de la même manière, le répertoire lib\[votre compilateur] doit aussi être accessible depuis votre compilateur. L'édition de liens se fera simplement en liant le fichier libirrKlang.a / irrKlang.lib. On remarquera que les bibliothèques statiques sont très légères, elles ne contiennent que les prototypes des fonctions qui, elles, sont dans la bibliothèque dynamique irrKlang.so / irrKlang.dll. Les so/dll devront être accessibles depuis votre exécutable, placez-les donc soit dans le même répertoire que votre exécutable, soit dans un répertoire du PATH de votre système.

2. Utilisation d'IrrKlang

Les plus attentifs d'entre vous l'auront remarqué, le nom IrrKlang est proche de Irrlicht, ce n'est pas anodin, la même personne a programmé ces deux outils et ils sont faits dans la même logique. C'est pourquoi IrrKlang est logiquement la bibliothèque à utiliser pour ajouter une gestion sonore dans un environnement Irrlicht. Les mêmes conventions sont utilisées, les mêmes espaces de noms sont utilisés, la gestion des fichiers est commune, bref, IrrKlang est pensé pour être utilisé conjointement avec Irrlicht. Vous pouvez néanmoins l'utiliser sans utiliser Irrlicht.

Tous les objets et toutes les méthodes propres à IrrKlang sont situés dans l'espace de noms irr::audio. C'est notamment le cas des deux objets de base : le moteur de sons et l'objet son. Il s'agit des classes ISoundEngine et ISound. Ces deux classes sont suffisantes pour réaliser la majeure partie de ce qu'on peut attendre d'une bibliothèque de gestions de sons.

Le moteur de sons va jouer les sons dans un espace 2D ou 3D, il va gérer le rendu de tous les sons, alors que les sons eux-mêmes vont se gérer indépendamment des autres sons.

IrrKlang est extrêmement simple à utiliser, une poignée de fonctions permettent d'accéder à toutes les fonctionnalités. Les fonctions utiles sont les suivantes :

- créer un moteur de sons
- jouer un son en 2D
- jouer un son en 3D
- sélectionner l'emplacement de l'utilisateur dans l'univers 3D
- éventuellement spécifier quelques options pour les sons

Voilà, ces quelques fonctionnalités résument presque entièrement IrrKlang. La création d'un moteur de sons se fait de la manière suivante :

```
ISoundEngine* engine = createIrrKlangDevice();
```

Vous pouvez aussi vous pencher sur les différents paramètres facultatifs de createIrrKlangDevice : vous pouvez spécifier le type de driver à utiliser ou bien des options sur le rendu des sons. Comme à son habitude dans Irrlicht, il conviendra de vérifier que le moteur a bien été créé en testant engine. Pour jouer un son, rien de plus simple :

```
ISound* music = engine->play2D("test.ogg");
```

De même, vous pouvez spécifier quelques options sur le rendu du son via les paramètres facultatifs de play2D. La lecture de sons en 3D n'est guère plus complexe, il suffit juste de rajouter la position 3D de la source sonore :

```
ISound* music = engine->play3D("test.ogg",  
core::vector3df(1,2,3));
```

Pour pouvoir jouer un son dans un espace 3D, il faut aussi indiquer la position et l'orientation de l'utilisateur :

```
engine->setListenerPosition(core::vector3df(1,0,2),  
core::vector3df(1,0,0));
```

On remarquera l'utilisation de la classe irr::core::vector3df de Irrlicht pour toutes les localisations spatiales.

Les fonctions play2D et play3D renvoient un ISound*. Il n'est nécessaire de récupérer ce pointeur que si vous souhaitez par la suite manipuler directement le son pour, par exemple, en changer manuellement le volume ou la balance. Il ne sera donc pas absurde de ne pas récupérer ce pointeur pour laisser le moteur gérer lui-même tout le rendu des sons.

Le rendu 2D est très simple, un son sera tout le temps entendu avec le même volume. Dans un univers 3D le moteur de sons va déterminer le volume (sur chaque canal) en fonction de la distance séparant la source de l'utilisateur et en fonction du type d'interpolation à effectuer. Chaque son est joué à son volume maximal à proximité de la source et est audible jusqu'à une distance maximale. Entre ces distances minimales et maximales, le son sera interpolé.

3. Fonctionnalités absentes

D'autres fonctionnalités sont présentes pour pouvoir spécifier des emplacements mémoires comme flux à lire ou encore pour utiliser votre propre chargeur de flux. Ces fonctionnalités restent cependant très anecdotiques.

On remarquera aussi la présence d'une bibliothèque dynamique ikpMP3.dll / so. Il s'agit du décodeur MP3 d'IrrKlang. Ce fichier doit être accessible depuis votre exécutable si vous souhaitez lire des mp3, dans le cas contraire, vous pourrez tout à fait vous en passer.

La lecture de fichiers mp3 s'avère légèrement plus gourmande que la lecture des fichiers ogg. Est-ce dû à l'implémentation du décodeur mp3 ? au format mp3 en lui-même ? sûrement un peu des deux.

2.1. Interactions avec Irrlicht

Les interactions avec Irrlicht sont très (trop?) faibles. Dans le couplage d'un moteur audio avec un moteur 3D, on pourrait espérer voir apparaître des fonctions pour calculer le rendu d'un son dans un univers 3D connu par Irrlicht, pour par exemple tenir compte des murs ou des matériaux. Mais ceci est également fortement remis en cause par l'absence d'analyse fréquentielle.

La seule interaction intéressante avec Irrlicht est la création des noeuds de scène "source sonore" et "écouteur", écrits par la communauté IrrKlang. Il suffit alors d'ajouter ces noeuds au graphe de scène Irrlicht pour avoir un rendu sonore. Et la cerise sur le gâteau est qu'on peut directement spécifier la caméra comme écouteur. Il s'agit de deux nouvelles classes : CSoundEmitterSceneNode et CSoundListenerSceneNode, qui héritent toutes deux de ISceneNode.

Le code source de ces deux classes est disponible sur le forum de la communauté IrrKlang ([Lien112](#)). Ca vaut le détour.

On remarque tout de suite l'absence d'analyse fréquentielle des sons, ce qui empêche la création d'effets sympathiques ou les modifications de l'equalizer. De même on ne peut pas paramétrer la qualité du sons en sortie, le son joué sera exactement celui du fichier spécifié.

On remarque aussi le faible nombre de formats supportés, mais est-ce vraiment un handicap ? Le wav, le mp3 et le ogg sont présents, les autres sont-ils vraiment nécessaires ?

Ce qui manque et qui se fait bien remarquer, c'est l'absence de callbacks sur événements à la manière de ceux présents dans FmodEx. Il n'est ainsi pas possible de déterminer quand un son se termine, on est obligé de scruter régulièrement l'état du son par la méthode isFinished(). Ce serait une fonctionnalité très intéressante qui pourrait être rajoutée dans les prochaines versions d'IrrKlang.

4. Conclusion

On est tout de suite séduit par IrrKlang et par son étonnante simplicité. Ce moteur aura tout naturellement une bonne place pour épauler Irrlicht. Actuellement, IrrKlang n'est pas encore open source, mais ça ne saurait tarder et il y a aussi de fortes chances qu'on le retrouve diffusé sous licence zlib. Ce serait un avantage incontestable face à son rival FmodEx, dont l'utilisation peut être onéreuse dans certains cas.

Cette bibliothèque pourra très vite devenir une valeur sûre pour tous ceux qui veulent gérer simplement des sons sans pour autant chercher les réglages fins ni entièrement réalistes.

Attendons donc la version 1.0 et la diffusion open source pour avoir une vraie bibliothèque utilisable, compétitive et qui n'aura pas à rougir devant sa principale rivale FmodEx.

Retrouvez l'article de Pierre Schwartz en ligne : [Lien113](#)

Les livres 2D/3D/Jeux

Ultimate 3D Game Engine Design & Architecture

Ultimate 3D Game Engine Design and Architecture is a complete reference for designing and creating a game engine from the ground up. The book covers the various systems and processes that go into a complete game engine, with an emphasis on the issues to consider when designing the architecture for the engine. Unlike other books that only focus on one aspect of the game engine, such as graphics or physics, this book focuses on a complete game engine from a cross-platform perspective. The depth of coverage this book provides gives programmers aspiring to get into game development and experienced game developers all the details they need to create a complete game engine, a sample of which is on the companion CD-ROM.

Critique du livre par la rédaction (Miles)

Après avoir lu ce livre, j'ai vraiment une impression très mitigée. Il a des côtés très biens et d'autres vraiment très mauvais. Il présente le moteur de jeu vidéo de l'auteur, multi-plateforme, ... Commençons par les mauvais côtés.

Ce livre ne présente pas la hiérarchie du moteur, la seule chose qu'on voit, c'est des cases représentant les classes, rien d'autres.

On parle d'UML, mais cela n'a d'UML que le nom... Ensuite, la qualité du code est... déplorable, vraiment. Pas de const-correctness, on utilise des char* pour les paramètres qui sont transformés en std::string juste après !! Du grand n'importe quoi. Dans le même genre, on crée des classes avant de les utiliser alors que le livre indique de ne pas le faire ! Enfin, ce livre sacrifie à la tradition, du code sur plusieurs pages parfois, alors qu'il est présent sur le CD et sur le site Internet, pourquoi gaspiller des pages pour ça ?

Bon, dans la partie des bonnes choses, il y a le fait que ce livre couvre presque toutes les parties d'un moteur de jeu vidéo, et multi-plateforme. Des entrées à la physique, en passant par le graphisme - 3D avec OpenGL ou DirectX - et l'IA, la gestion de ces systèmes est exposée. Si certaines parties présentent une abstraction d'autres bibliothèques interchangeables - comme pour le graphisme -, ce n'est pas le cas pour toutes, comme la physique.

En parlant de la gestion de la physique, elle dispose d'une bonne partie du livre pour exposer son fonctionnement et sa manière de fonctionner, même si elle est relativement limitée, pour l'instant.

On regrettera que l'exposé se fasse sur une version 0.5 d'un moteur en évolution. On regrettera aussi l'exposé du fonctionnement du scripting, sachant que c'est légèrement faux -

compilé ou interprété, il peut toujours s'agir d'une machine virtuelle pour un langage de script -, et qu'on aurait pu prendre des pages pour mieux présenter l'architecture du moteur...

Pour conclure, il faut voir que ce livre est, à ma connaissance, le seul livre complet sur un moteur de jeu vidéo. Il existe des livres plus complets sur ces sous-parties - surtout moteur 3D -, mais rien sur la totalité. Allen Sherod est aussi l'auteur d'un livre sur les structures de données orientées jeu qui va être mis à jour en mars 2007, il devrait être du même niveau que ce livre-ci.

Retrouvez ce livre sur la rubrique Jeux : [Lien114](#)

Open GL 2.0 : Guide officiel

OpenGL est une interface logicielle puissante qui permet de produire des images haute qualité générées par ordinateur et des applications interactives grâce à des objets bidimensionnels et tridimensionnels, des bitmaps et des images en couleur. Cet ouvrage vous livre des informations complètes sur OpenGL et Open GL Utility Library. Cette édition décrit les toutes dernières fonctionnalités des versions 1.5 et 2.0 d'OpenGL, et vous introduit également au GLSL (OpenGL Shading Language). Vous y trouverez des explications claires sur la fonctionnalité d'OpenGL et de nombreuses techniques de programmation infographique telles que la construction et la restitution de modèles 3D, la visualisation interactive des objets depuis différents points de vue, l'utilisation des dégradés et les effets d'éclairage et de texture permettant d'améliorer le réalisme des scènes. De plus, ce livre approfondit des techniques avancées comme l'application de texture, le lissage, le fog et autres effets atmosphériques, les NURBS, le traitement d'images, etc. Au fil de ce texte clair et détaillé, vous trouverez aussi des remarques relatives à l'optimisation des performances, aux extensions OpenGL et aux techniques inter-plates-formes.

Critique du livre par la rédaction (Edouard Kaiser)

Ce livre de plus de 700 pages est certainement le livre de chevet idéal pour tous les programmeurs de l'API OpenGL. Qu'ils soient débutants ou expérimentés chacun trouvera dans cette édition ce qu'il y a à savoir sur l'API dans sa version 2.0. En partant des bases pour finir sur les dernières fonctionnalités telles que le GLSL, ce livre aborde tous les domaines (Nurbs, Texturing, Blending, Picking, Eclairage, etc.) de manière simple et très explicite en s'appuyant sur des parties de codes et schémas pour alimenter la théorie. Vous saurez tous sur les fonctions composant cette API et vous en apprendrez certainement des nouvelles ! On peut facilement qualifier ce livre de bible francophone et il le mérite amplement. Le seul bémol en ce qui me concerne est le code qui s'appuie souvent sur la bibliothèque GLUT. On aurait très certainement préféré des exemples sur SDL qui est beaucoup plus d'actualité et soutenu que GLUT, mais ce n'est qu'un détail et cela n'enlève en rien la qualité irréfutable de cette oeuvre.

Retrouvez ce livre sur la rubrique Jeux : [Lien115](#)

La 3D libre avec Blender

Présentation de l'éditeur

Logiciel libre de référence pour le graphisme et l'animation 3D, tant sous MS-Windows et Linux que sous Mac OS X, Blender 2.42 défie les ténors du domaine tels XSI, 3ds Max, Maya... Grâce à son interface intelligente et contextuelle, il permet à l'artiste d'allier productivité et créativité. Très ouvert, il met à sa

disposition de nombreux moteurs de rendu photoréalistes tel YaFray et s'intègre, par sa richesse et sa flexibilité, à la panoplie des outils de travail quotidiens du graphiste professionnel.

Biographie de l'auteur

Utilisateur de Blender depuis près de sept ans, Olivier Saraja est l'un des animateurs de la communauté du graphisme libre - en particulier sous Linux - et est l'auteur de nombreux didacticiels pour Blender, POV-Ray, KpovModeler ou Art of Illusion. Il a contribué à la documentation officielle de Blender et oeuvre pour en faciliter l'accès au plus grand nombre, dans les communautés françaises comme anglophones.

Critique du livre par la rédaction (Olivier Van Hoof)

Cet excellent livre, réalisé par un passionné de Blender, est le guide indispensable à tout qui veut apprendre à se servir de ce fabuleux logiciel libre de création 3D qu'est Blender. Notons qu'il peut servir à la fois au débutant complet en 3D ainsi qu'aux personnes ayant déjà des connaissances en la matière, vu que des comparaisons avec les ténors du marché que sont 3DStudio et Maya sont évoquées tout au long de l'ouvrage.

Après le traditionnel chapitre sur l'installation (quasiment inévitable dans tout ouvrage informatique), on entre dans le vif du sujet avec une présentation de l'interface utilisateur, qu'il est essentiel d'apprendre à connaître pour pouvoir jongler facilement entre les différentes vues graphiques en cours de travail, ainsi que se repérer dans la scène 3D. S'en suit un premier tutoriel pour découvrir les notions de base du logiciel, ce qui donne irrésistiblement envie d'aller plus loin.

Le chapitre suivant traite de la modélisation 3D de manière plus détaillée, en n'abordant toutefois pas les objets de type NURBS, ce qui est un choix volontaire de l'auteur et judicieux car ce type d'objets sera totalement revu dans les nouvelles versions de Blender, et qu'il est tout à fait possible d'utiliser à la place les courbes de Bézier, qui sont également plus faciles à appréhender. Certaines techniques avancées comme les méthodes de déformation ou la génération de particules ne sont bizarrement présentées que dans la partie animation, quelques chapitres plus loin, pourtant elles servent aussi pour une image statique.

L'habillage des squelettes 3D en leur créant des matériaux réalistes est une des étapes-clé d'une belle image, et est bien traitée dans le chapitre suivant. On peut cependant regretter l'absence de quelques "recettes de cuisine" simples pour illustrer la réalisation de matériaux courants comme l'or, le métal, le bois, le marbre, le verre, etc.

Les chapitres suivants traitent de l'éclairage, fondamental pour obtenir une image de qualité, ainsi que de l'animation et des techniques avancées (modélisation de fluides, de corps souples, etc.) qui montrent l'étendue impressionnante des possibilités de Blender. Enfin, la phase de rendu et les effets de post-production qu'il permet (notamment la profondeur de champ parfaitement paramétrable) sont expliqués.

Au final, même s'il n'est pas une référence exhaustive de Blender, ce qu'il ne se prétend d'ailleurs pas, cet ouvrage est un excellent guide pour apprendre à créer des scènes 3D avec Blender, et qui plus est l'un des seuls en français. Au terme de la lecture, le niveau de connaissance acquis est très bon. A mettre donc entre toutes les mains des artistes 3D francophones :)

Retrouvez ce livre sur la rubrique Jeux : [Lien116](#)

Liens

- Lien1 : <http://www.worldofjava.org/homepage.do>
Lien2 : <http://baptiste-wicht.developpez.com/tutoriel/java/woj/>
Lien3 : <http://www.developpez.net/forums/forumdisplay.php?f=590>
Lien4 : <http://blog.developpez.com/index.php?blog=124&p=2812&more=1&c=1&tb=1&pb=1>
Lien5 : <http://java.developpez.com/cours/#eclipse>
Lien6 : <http://www.developpez.net/forums/forumdisplay.php?f=94>
Lien7 : http://blog.developpez.com/index.php?blog=12&title=forums_plugins_eclipse_et_applications_r&more=1&c=1&tb=1&pb=1
Lien8 : http://www.theserverside.com/news/thread.tss?thread_id=44775&asrc=EM_NLN_1211787&uid=5780676
Lien9 : http://www.java.com/fr/download/help/cache_virus.xml
Lien10 : http://www.f-secure.com/v-descs/openstream_t.shtml
Lien11 : http://blog.developpez.com/index.php?blog=127&title=java_victime_de_1_antivirus&more=1&c=1&tb=1&pb=1
Lien12 : <http://www.developpez.net/forums/showthread.php?t=167330>
Lien13 : <http://www.bejug.org/>
Lien14 : <https://dfjug.dev.java.net/>
Lien15 : <http://java.sun.com/javaone/sf/>
Lien16 : <http://www.developpez.net/forums/calendar.php?do=getinfo&e=134&day=2007-6-27&c=1>
Lien17 : <http://www.springone.com/>
Lien18 : http://blog.developpez.com/index.php?blog=12&title=les_avantages_du_jug&more=1&c=1&tb=1&pb=1
Lien19 : <http://java.developpez.com/livres/?page=Francais#L2212120192>
Lien20 : http://java.developpez.com/faq/gui/?page=graphique_general_dessin#GRAPHIQUE_DESSIN_texte
Lien21 : http://java.developpez.com/faq/gui/?page=graphique_general_dessin#GRAPHIQUE_doublebuffering
Lien22 : <http://www.developpez.net/forums/forumdisplay.php?f=629>
Lien23 : <http://www.developpez.net/forums/forumdisplay.php?f=630>
Lien24 : <http://www.developpez.net/forums/forumdisplay.php?f=631>
Lien25 : <http://java.developpez.com/faq/maven/?page=terminologie#documentation1>
Lien26 : <http://maven.apache.org/continuum/>
Lien27 : <http://java.developpez.com/livres/?page=Francais#L2100506234>
Lien28 : http://blog.developpez.com/index.php?blog=58&title=installer_beryl_sous_xfce4&more=1&c=1&tb=1&pb=1
Lien29 : <http://linux.developpez.com/livres/?page=livresSYS#L2841772438>
Lien30 : <http://linux.developpez.com/livres/?page=livresDIST#L2746033739>
Lien31 : <http://carteyron.chez-alice.fr/mvu/modelixe.php>
Lien32 : <http://phplib.sourceforge.net/>
Lien33 : <http://www.phpbb.com/>
Lien34 : <http://smarty.php.net/>
Lien35 : <http://www.tinybutstrong.com/fr/>
Lien36 : <http://www.developpez.net/forums/member.php?u=16619>
Lien37 : <http://pear.php.net/manual/fr/standards.php>
Lien38 : <http://www.gnu.org/copyleft/gpl.html>
Lien39 : http://fr.wikipedia.org/wiki/Licence_publique_g%C3%A9n%C3%A9rale_GNU
Lien40 : <http://www.gnu.org/licenses/lgpl.html>
Lien41 : <http://fr.wikipedia.org/wiki/LGPL>
Lien42 : <http://phpsavant.com/yawiki/index.php?area=Savant3>
Lien43 : <http://yats.sourceforge.net/>
Lien44 : <http://cyberzoide.developpez.com/php4/template/>
Lien45 : http://genova.developpez.com/articles/template_phpbb/
Lien46 : <http://php.developpez.com/cours/#bibliotheques-gabarits>
Lien47 : <http://www.developpez.net/forums/showthread.php?t=3223>
Lien48 : <http://g-rossolini.developpez.com/comparatifs/php/templates/>
Lien49 : http://blog.developpez.com/index.php?blog=102&title=les_dernieres_actualitesZend&more=1&c=1&tb=1&pb=1
Lien50 : <http://vincentlaine.developpez.com/tuto/dotnet/wcf/>
Lien51 : <http://c.developpez.com/cours/#cours-c>
Lien52 : http://mapage.noos.fr/emdel/init_c.htm
Lien53 : <http://mapage.noos.fr/emdel/codage.htm>
Lien54 : <http://c.developpez.com/livres/#L2100490184>
Lien55 : <http://www.codeblocks.org/>
Lien56 : http://sourceforge.net/project/downloading.php?groupname=codeblocks&filename=codeblocks1.0rc2_mingw.exe&use_mirror=ovh
Lien57 : <http://forums.codeblocks.org/index.php/board.20.0.html>
Lien58 : <http://fearyourself.developpez.com/tutoriel/linux/ubuntu/installation-code-blocks/>
Lien59 : http://prdownload.berlios.de/codeblocks/CB_20061220_rev3402_macppc.zip
Lien60 : http://prdownload.berlios.de/codeblocks/CB_20070301_rev3656_macx86.zip
Lien61 : <http://loulou.developpez.com/tutoriels/cpp/codeblocks/>
Lien62 : <http://c.developpez.com/cours/#tutos-c>
Lien63 : <http://mapage.noos.fr/emdel/notes.htm>
Lien64 : <http://mapage.noos.fr/emdel/comprog.htm>
Lien65 : <http://mapage.noos.fr/emdel/tad.htm>
Lien66 : <http://mapage.noos.fr/emdel/reseaux.htm>
Lien67 : <http://mapage.noos.fr/emdel/pthreads.htm>
Lien68 : <http://mapage.noos.fr/emdel/inputs.htm>
Lien69 : <http://c.developpez.com/livres/#L2100487345>
Lien70 : http://www.amazon.fr/Exercices-corrige%C3%A9s-sur-langage-Solutions/dp/2100050311/ref=pd_bxgy_b_img_b/402-3738394-1135358
Lien71 : <http://c.developpez.com/livres/#L2841772896>
Lien72 : <http://c.developpez.com/livres/#L2841770966>
Lien73 : <http://c.developpez.com/livres/#L2212116012>
Lien74 : <http://www.recreomath.qc.ca/index.htm>
Lien75 : http://www.topdevone.com/index_fr.php
Lien76 : <http://www.france-ioi.org/>
Lien77 : <http://www.developpez.net/forums/forumdisplay.php?f=18>

Lien79 : <http://www.developpez.net/forums/forumdisplay.php?f=373>
Lien80 : <http://franckh.developpez.com/articles/c-ansi/bien-debuter-en-c>
Lien81 : <http://www.ietf.org/rfc/rfc0822.txt>
Lien82 : <http://jlr31130.free.fr/rfc822.html>
Lien83 : <http://broux.developpez.com/articles/protocoles/smtp/>
Lien84 : <http://c.developpez.com/livres/#L2841774074>
Lien85 : <http://c.developpez.com/livres/#L274298089X>
Lien86 : <http://www.w3.org/P3P/validator.html>

Lien88 : <http://xml.developpez.com/livres/#L2212116756>
Lien89 : <http://xml.developpez.com/livres/#L2841773531>
Lien90 : <http://dolphy35.developpez.com/article/access2007/FormContinus/>
Lien91 : <http://warin.developpez.com/access/ruban/>
Lien92 : <http://warin.developpez.com/access/favoris>
Lien93 : <http://warin.developpez.com/access/piecejointe>
Lien94 : <http://warin.developpez.com/access/pdf>
Lien95 : <http://warin.developpez.com/access/nouveautes2007>
Lien96 : <http://www.mozilla-europe.org/fr/products/thunderbird>
Lien97 : <http://www.mozilla.org/products/thunderbird/dictionaries.html>
Lien98 : <https://addons.mozilla.org/thunderbird/themes/>
Lien99 : <https://addons.mozilla.org/thunderbird/extensions/>
Lien100 : <http://jcrozier.developpez.com/articles/logiciels/thunderbird>
Lien101 : <http://whiler.developpez.com/tutoriels/borland/codegear/bds/delphi/ota/>
Lien102 : <http://delphi.developpez.com/defi/>
Lien103 : <http://delphi.developpez.com/defi/puissance4/topic/>
Lien104 : <http://delphi.developpez.com/defi/puissance4/>
Lien105 : <http://delphi.developpez.com/defi/puissance4/vainqueur/>
Lien106 : <http://www.ruby-lang.org/fr/>
Lien107 : <http://pbnaigeon.developpez.com/tutoriel/ruby/premiers-pas-avec-ruby/>
Lien108 : <http://ruby.developpez.com/livres/#L2841773884>
Lien109 : <http://ruby.developpez.com/faq>
Lien110 : <http://baptiste-wicht.developpez.com/tutoriel/conception/mvc/>
Lien111 : <http://www.ambiera.com/irrklang/index.html>
Lien112 : <http://irrlight.sourceforge.net/phpBB2/viewtopic.php?t=17287>
Lien113 : <http://khayam.developpez.com/articles/cpp/irrklang/>
Lien114 : <http://jeux.developpez.com/livres/#L1584504730>
Lien115 : <http://jeux.developpez.com/livres/#L2744020869>
Lien116 : <http://jeux.developpez.com/livres/#L2212119593>
Lien117 : <http://g-rossolini.developpez.com/comparatifs/php/templates/?page=conclusion#L.VII-A-2>
Lien118 : <http://delphi.developpez.com/defi/puissance4/#L.IV>
Lien119 : <http://delphi.developpez.com/defi/serpent/>