

Developepez

Hors série

Le Mag

Hors série spécial rentrée.
HS 2.

Magazine en ligne gratuit.

Diffusion de copies conformes à l'original autorisée.

Réalisation : Alexandre Pottiez

Rédaction : la rédaction de Developepez

Contact : magazine@redaction-developpez.com

Sommaire

Java	Page 2
Android	Page 5
Eclipse	Page 10
Outlook	Page 15
Word	Page 17
PowerPoint	Page 31
Excel	Page 39
Access	Page 50
C & C++	Page 59
Qt	Page 65
Web sémantique	Page 71
Business Intelligence	Page 77
Solutions d'entreprise	Page 80
Pascal	Page 82
Perl	Page 88
2D/3D/jeux	Page 90
Liens	Page 103

Article Word



Fusion et publipostage, par l'interface et en VBA

Word peut se révéler un outil de choix pour la diffusion en masse de documents personnalisés. Écrire une seule lettre qui sera différente pour chaque destinataire.

par **Olivier Lebeau**
Page 17

Article Qt



Débuter dans la création d'interfaces graphiques avec Qt 4

Ce tutoriel fournit une introduction à la programmation d'interfaces graphiques utilisant la boîte à outils Qt.

Traduit par **l'équipe Qt de Devlopez.com**
Page 65

Éditorial

Dans ce numéro spécial, vous retrouverez, sur plus d'une centaine de pages, nos meilleurs articles pour bien débuter dans près de seize technologies traitées dans ce numéro et que vous êtes susceptible d'étudier cette rentrée.

Bonne lecture.

Profitez-en bien !

La rédaction

Bien débiter en Java

Vous voulez commencer à programmer en Java et vous êtes perdu, vous ne savez pas par quoi commencer ? Alors cet article est fait pour vous. Ce n'est pas un tutoriel du langage Java, c'est plutôt un guide qui va vous aider à débiter, à choisir vos outils et vous indiquer quelles ressources vous pourriez lire pour avancer.

1. Qu'est-ce que le Java ?

Pour commencer, Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton de Sun Microsystems. Mais c'est également un environnement d'exécution.

Java peut être séparé en deux parties. D'une part, votre programme écrit en langage Java et d'autre part, une machine virtuelle (JVM) qui va se charger de l'exécution de votre programme Java.

C'est cette plateforme qui garantit la portabilité de Java. Il suffit qu'un système ait une machine virtuelle Java pour que tout programme écrit en Java puisse fonctionner.

Avec le langage Java, vous pouvez développer, des applications Desktop, développer des applets pour vos sites web, développer des sites en JSP ([Lien 01](#)), des applications pour téléphone mobile. La première chose à faire est bien évidemment d'apprendre à faire des applications stand alone simples.

2. Je débute

Il n'est pas facile de commencer à développer si on n'a jamais fait de programmation auparavant. C'est pourquoi, il est très important de commencer avec des bonnes ressources et de ne pas se lancer à la va-vite dans le langage.

Pour les personnes connaissant déjà la programmation et les langages objet, l'apprentissage de Java ne devrait pas se révéler trop compliqué.

Vous pouvez commencer par lire l'un des tutoriels ([Lien 02](#)) d'initiation à Java de la rubrique Java ([Lien 03](#)) de Developpez.com. Je vous conseille d'en lire au moins un parmi ceux-là ou alors un des livres proposés :

- Penser en Java V2 : [Lien 04](#)
- Développez en Java : [Lien 05](#)
- Cours sur le langage Java : [Lien 06](#)
- Le livre de Java premier langage : [Lien 07](#)
- Programmer en Java : [Lien 08](#)
- Java en concentré : [Lien 09](#)

Il vous faudra bien évidemment un ou des outils pour développer en Java. Néanmoins pour commencer, je vous conseille d'utiliser un simple bloc-note et, de compiler et d'exécuter vos programmes en ligne de commande pour commencer. Cela vous permettra de bien comprendre le fonctionnement de Java. Ensuite, vous pourrez bien sûr développer avec un EDI plus évolué : [Lien 10](#).

Je vous conseille d'utiliser Eclipse ([Lien 11](#)) ou Netbeans ([Lien 12](#)) pour développer vos applications Java. Tous deux sont multiplateformes, gratuits et très puissants.

3. Je veux aller plus loin

Le langage Java peut être utilisé dans de nombreux domaines que je vous invite à découvrir en parcourant les différents articles Java de Developpez.com. Les chapitres suivants reprennent quelques thèmes que vous pourriez approfondir.

3.1. Comment développer des interfaces graphiques avec Java ?

Pour développer des interfaces graphiques en Java, plusieurs choix s'offrent à vous :

- **AWT** : c'est le premier framework graphique de Java. Il utilise les composants système pour dessiner l'interface graphique, il est donc limité en composants : [Lien 13](#) ;
- **Swing** : c'est le successeur de AWT. Tous les composants sont dessinés directement en Java. Ce qui fait qu'il est plus lent que AWT, mais cette différence est très faible dans les dernières versions de Java : [Lien 14](#) ;
- **SWT** : ce framework graphique n'est pas disponible avec l'API standard de Java. C'est l'équipe de développement d'Eclipse qui l'a créé. Comme AWT, il utilise directement les composants système mais émule les composants non disponibles en les dessinant lui-même.

Je vous conseille d'utiliser Swing. En effet ce framework est très simple d'utilisation et très complet. De plus, la communauté autour du framework est énorme et permet d'ajouter de nombreuses fonctionnalités à Swing. Vous pouvez aussi tester SWT pour vous faire une idée.

Voici de quoi débiter avec ces frameworks :

- Création de votre première interface graphique avec Swing : [Lien 15](#) ;
- Les principes de base du GUI avec Swing : [Lien 16](#) ;
- Le développement d'interfaces graphiques avec SWT : [Lien 17](#) ;
- Le livre "Swing la synthèse" : [Lien 18](#) ;
- Le livre "Les cahiers du programmeur Swing" : [Lien 19](#).

3.2. Comment gérer les bases de données avec Java ?

La première manière de gérer les bases de données est d'utiliser l'API JDBC qui est en standard avec Java ([Lien 20](#)). Ensuite, pour aller plus loin, vous pouvez vous orienter vers les frameworks de mapping objet/relationnel tels que Hibernate ([Lien 21](#)) ou encore utiliser un framework de persistance plus petit tel qu'iBatis.

Voici des ressources qui vous seront fort utiles pour votre apprentissage :

- Comment utiliser un pilote de base de données JDBC : [Lien 22](#) ;
- Tutoriel JDBC : [Lien 23](#) ;
- La persistance des données avec Hibernate 2.1.8 : [Lien 24](#) ;
- La persistance en base de données avec JPA (Java Persistence API) : [Lien 25](#) ;
- Le livre "Hibernate 3.0 - Gestion optimale de la persistance dans les applications Java/J2EE" : [Lien 26](#) ;
- Le site officiel d'iBatis : [Lien 27](#).

3.3. Comment faire de la programmation web avec Java ?

La programmation Java EE ([Lien 28](#)) est un thème très large. En effet, il existe une grande quantité de frameworks web ainsi que différentes manières de programmer pour le web (JSP ([Lien 29](#)), Servlets ([Lien 30](#)), Applets ([Lien 31](#))) et de nombreux outils (serveur d'application (JBoss, Weblogic) ou conteneur Web Java (Tomcat par exemple : [Lien 32](#))). Je vous conseille donc de bien étudier le sujet avant de vous y lancer.

Voici quelques ressources qui pourraient vous être utiles :

- Java Standard Tag Library (JSTL) : [Lien 33](#) ;
- Introduction à Struts : [Lien 34](#) ;
- Programmation web avec Java : [Lien 34](#) ;
- Les cahiers du programmeur Java, conception et déploiement J2EE : [Lien 35](#) ;
- Struts par la pratique : [Lien 36](#).

3.4. Comment développer pour un mobile avec Java ?

Il existe peu de documentation sur le développement Java ME : [Lien 37](#). Vous pouvez tout de même lire cet article : Débuter en J2ME avec le profil MIDP ([Lien 38](#)) ou alors vous pouvez consulter la documentation de Sun qui est très complète mais n'existe qu'en anglais ([Lien 39](#)).

3.5. Comment faire de la 2D/3D en Java ?

En Java, vous pouvez aussi faire du dessin (graphisme) 2D et 3D. Vous pouvez utiliser différentes API pour faire ceci : Java2D, Java3D, JOGL, LWJGL et encore quelques autres moteurs graphiques. Voici quelques ressources qui vous aideront à démarrer avec le graphisme :

- Introduction au dessin en Java : [Lien 40](#) ;
- La programmation OpenGL en Java : introduction à JOGL : [Lien 41](#) ;
- Premiers pas avec la librairie Java 3D : [Lien 42](#) ;
- Les API 3D pour Java : [Lien 43](#).

3.6. Quels sont les autres outils que je peux utiliser ?

Une fois passé le stade des premiers programmes, il sera utile de voir d'autres programmes utilitaires pour Java et de tester de nouveaux EDI. Découvrez les EDI les plus populaires sur notre page outils : [Lien 44](#).

Vous pourrez aussi être amené à devoir créer un exécutable de votre application. Pour cela, retrouvez toutes les techniques possibles dans cet article : [Lien 45](#).

Pour la gestion de vos builds, où vous pourriez aussi être amené à utiliser des utilitaires dédiés, vous pouvez vous diriger vers Ant ([Lien 46](#)) ou Maven ([Lien 47](#)).

4. Comment tester mes connaissances ?

Faire une série d'exercices pour tester vos connaissances peut se révéler très intéressant. En effet, ceci vous permettra de voir si vous avez bien acquis les informations sur Java.

C'est pourquoi l'équipe Java vous propose une série de petits exercices sur le langage Java : Série d'exercices en Java ([Lien 48](#)).

Une autre façon de s'entraîner est de créer une petite application soi-même. Il ne faut pas viser trop gros sans quoi vous risqueriez de ne jamais arriver au bout de votre application. Voici quelques petits exemples d'applications à développer soi-même qui peuvent se révéler intéressantes :

- **Une calculatrice** : pour commencer, développer une calculatrice se révèle assez simple, mais vous pouvez aussi aller assez loin suivant les fonctionnalités que vous lui ajoutez ;
- **Un petit jeu** : vous pouvez aussi développer un petit jeu, par exemple un puissance 4 ou un jeu d'échec. Vous n'avez pas encore besoin de travailler profondément sur le graphisme, mais vous pouvez déjà faire de belles choses. Si vous voulez aller un peu plus loin, vous pouvez vous lancer dans un jeu de la vie ;
- **Une bibliothèque** : vous pouvez également développer une petite bibliothèque. Cela vous permettra d'aborder les bases de données ainsi que les formulaires.

Bonne chance !

5. J'ai tout lu, mais pas tout compris, où poser mes questions ?

Il est possible que vous n'ayez pas compris une notion de Java ou que vous ayez un problème pour implémenter quelque chose. La première chose à faire est de vérifier dans les tutoriels Java si vous ne trouvez pas la réponse. Ensuite, vous pouvez consulter les différentes FAQ de la rubrique :

- FAQ Général Java : pour tout ce qui concerne le langage en général et les notions principales de Java : [Lien 49](#) ;
- FAQ Java GUI : pour tout ce qui concerne les interfaces graphiques en Java : [Lien 50](#) ;
- FAQ Java XML : pour tout ce qui concerne la manipulation de XML en Java : [Lien 51](#) ;

- FAQ Java EE : pour tout ce qui concerne Java EE : [Lien 52](#) ;
 - FAQ Java ME : pour tout ce qui concerne Java ME : [Lien 53](#) ;
 - FAQ JDBC : pour tout ce qui concerne la manipulation des bases de données : [Lien 54](#) ;
 - FAQ Hibernate : pour tout ce qui concerne Hibernate : [Lien 55](#) ;
 - FAQ Struts : pour tout ce qui concerne le framework struts : [Lien 56](#) ;
 - FAQ Eclipse : pour tout ce qui concerne Eclipse : [Lien 57](#) ;
 - FAQ Netbeans : pour tout ce qui concerne Netbeans : [Lien 58](#) ;
 - FAQ JCreator : pour tout ce qui concerne JCreator : [Lien 59](#) ;
 - FAQ Maven 2 : pour tout ce qui concerne Maven 2 : [Lien 60](#).
- Si votre réponse ne se trouve pas dans une de ces pages, vous pouvez poser votre question sur les forums Java :
- Général Java : pour toutes vos questions générales Java : [Lien 61](#) ;
 - Et ses sous-forums :
 - Langage : langage Java, syntaxe, POO, conventions : [Lien 62](#) ;
 - API : forum sur les API standard et API complémentaires : [Lien 63](#) ;
 - JDBC : pour vos questions sur l'API JDBC : [Lien 64](#) ;
 - Persistance : pour toutes vos questions sur la gestion de la persistance en Java : [Lien 65](#) ;
 - Spring : pour vos questions sur le framework Spring : [Lien 66](#) ;
 - XML : pour vos questions sur la manipulation des fichiers XML : [Lien 67](#) ;
 - Java ME : pour vos questions sur le développement pour mobiles : [Lien 68](#).
 - Interfaces graphiques en Java : pour toutes vos questions sur les interfaces graphiques en Java : [Lien 69](#) ;
 - Et ses sous-forums :
 - AWT/Swing : pour toutes vos questions sur les API Swing et AWT : [Lien 70](#) ;
 - SWT/JFace : pour toutes vos questions sur les API SWT et JFace : [Lien 71](#) ;
 - Graphisme : pour toutes vos questions sur le graphisme (2D, 3D, OpenGL...) en Java : [Lien 72](#).
 - Développement web avec Java : pour toutes vos questions sur la programmation web en Java : [Lien 73](#) ;
 - Serveurs d'applications pour Java/Java EE : pour toutes vos questions sur la programmation web en Java : [Lien 74](#) ;
 - EDI/Outils pour Java : pour toutes vos questions sur les différents outils pour Java : [Lien 75](#).

Lorsque vous posez une question, soyez toujours le plus précis possible et évitez les questions du type "Urgent problème db". Faites aussi attention à poster dans le bon sous-forum. Plus votre question sera claire et plus il y aura de gens qui la liront et prendront la peine d'essayer d'y répondre.

Et voilà, j'espère que vous trouverez les réponses à vos questions.

6. Conclusion

J'espère que cet article aura pu vous aider à vous lancer dans le langage Java plus facilement et que vous aurez acquis de nouvelles connaissances.

Retrouvez l'article de Baptiste Wicht en ligne : [Lien 76](#)



Introduction à la programmation sous Android

Ce tutoriel a pour but de vous présenter succinctement Android, ainsi que les prémices de la programmation sous celui-ci.

1. Android, c'est quoi ?

Android est un OS mobile Open Source pour smartphone, PDA, MP3 et tablette. Conçu initialement par Android Inc, il a été racheté par Google en 2005.

Pour commencer la programmation Android, il faut d'abord installer le **SDK Android** et comprendre les bases de la programmation sous Android. Puis nous allons faire notre premier programme sous Android c'est-à-dire le bien connu « **Hello Word** » pour bien comprendre ces bases.

2. Composantes d'une application Android

Une application Android est composée d'éléments de base :

2.1. Activities (Activités en français)

Une activité est la composante principale pour une application Android. Elle représente l'implémentation métier dans une application Android.

Prenant l'exemple d'une application qui liste tous vos fichiers mp3 présents dans votre téléphone, le projet pourrait se décomposer comme ci-dessous :

- une vue pour afficher la liste des mp3 ;
- une activité pour gérer le remplissage et l'affichage de la liste ;
- si l'on veut pouvoir rajouter, supprimer des mp3, on pourrait rajouter d'autres activités.

2.2. Services

Un service, à la différence d'une activité, ne possède pas de vue mais permet l'exécution d'un algorithme sur un temps indéfini. Il ne s'arrêtera que lorsque la tâche est finie ou que son exécution est arrêtée.

Il peut être lancé à différents moments :

- au démarrage du téléphone ;
- au moment d'un événement (arrivée d'un appel, SMS, mail, etc.) ;
- lancement de votre application ;
- action particulière dans votre application.

2.3. Broadcast and Intent Receivers

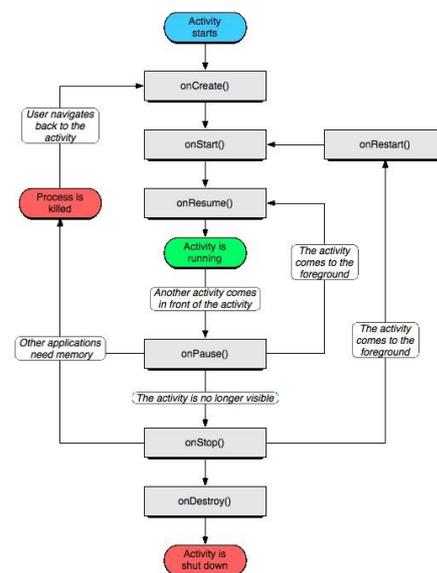
Un Broadcast Receiver comme son nom l'indique permet d'écouter ce qui se passe sur le système ou sur votre application et déclencher une action que vous aurez prédéfinie. C'est souvent par ce mécanisme que les services sont lancés.

2.4. Content providers

Les « content providers » servent à accéder à des données depuis votre application. Vous pouvez accéder :

- aux contacts stockés dans le téléphone ;
- à l'agenda ;
- aux photos ;
- ainsi qu'à d'autres données depuis votre application grâce aux content providers.

3. Cycle de vie d'une application Android



3.1. onCreate

Cette méthode est appelée à la création de votre activité (Activity). Elle sert à initialiser votre activité ainsi que toutes les données nécessaires à cette dernière.

Quand la méthode **onCreate** est appelée, on lui passe un Bundle en argument. Ce Bundle contient l'état de sauvegarde enregistré lors de la dernière exécution de votre activité.

3.2. onStart

Cette méthode est appelée dans le cas où votre application est en arrière-plan et qu'elle repasse en avant-plan.

Si votre activité ne peut pas aller en avant-plan quelle que soit la raison, l'activité sera transférée à **onStop**.

3.3. onResume

Cette méthode est appelée après **OnStart** (au moment où votre application repasse en avant-plan).

OnResume est aussi appelée quand votre application passe en arrière-plan à cause d'une autre application.

3.4. onPause

Appelée juste avant qu'une autre activité que la vôtre passe en **OnResume**. À ce stade, votre activité n'a plus accès à l'écran, vous devez arrêter de faire toute action en rapport avec l'interaction utilisateur. Vous pouvez par contre continuer à exécuter des algorithmes nécessaires mais qui ne consomment pas trop de CPU.

3.5. onStop

Appelée quand votre activité n'est plus visible quelle que soit la raison.

3.6. onDestroy

Appelée quand votre application est totalement fermée (Processus terminé).

4. Installer votre environnement de développement

4.1. Installation du SDK Android

1. Pour commencer allez sur le lien suivant : [Lien 77](#) et téléchargez la version du SDK qui convient à votre OS. Pour la suite on est sur un Windows 7.
2. Vous avez le choix entre la version Zip ou la version exe. Dans cet exemple on a pris la version Zip.
3. Une fois le SDK téléchargé, allez dans le dossier où se trouve le fichier Zip et l'extraire dans le dossier de votre choix.
4. Lancez l'exécutable « SDK Setup » qui se trouve à la racine du dossier.
5. La fenêtre suivante apparaît (la dernière version actuelle du SDK est 2.3) :



6. Choisissez ce que vous voulez installer. Par exemple :
 - **SDK Platform Android 2.x** : correspond tout simplement au SDK Android basique en version 2.x ;
 - **Samples for SDK API 7** : correspond à quelques exemples ;
 - **Android + Google APIs** : correspond au SDK Android (1re option) + Google Api qui inclut différentes fonctions comme GoogleMap, etc. ;
 - **Galaxy Tab** : c'est le SDK pour la tablette Samsung Galaxy Tab.
 - En cas de problème d'installation, allez dans « **Settings** » et cochez « **Force https:// sources to be fetched using http://** ».

7. Cliquez sur « Install ».

8. Installez Eclipse : [Lien 78](#) (téléchargez une version comprenant Java, soit la version pour les développeurs Java ou la version pour les développeurs J2EE) : version installée pour ce tutoriel : 3.6.1 Helios.

9. Il faut aussi installer JDK (Java Development Kit) et JRE (Java Runtime Environment) si ce n'est pas déjà fait : [Lien 79](#).

10. Lancez votre Eclipse, allez dans le menu « Help and Install New Software »

11. Dans la partie « Available Software », cliquez sur « Add ».

12. Rajoutez le nom du site (« ADT plugin » par exemple). Dans la location rajoutez : [Lien 80](#), puis cliquez sur OK.

13. Revenez dans « Available Software », vous devez voir « Developer Tools ». Sélectionnez-le et appuyez sur « Next ».

14. Cliquez sur « Next » pour lire et accepter la licence et puis cliquez sur « Finish ».

15. Pour finir l'installation relancez Eclipse.

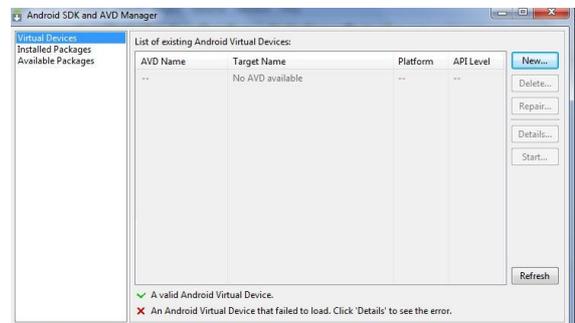
Voilà votre environnement de développement est prêt à être utilisé.

4.2. Configuration de votre environnement de développement

Vous avez dû remarquer qu'un nouvel élément est apparu dans votre Eclipse dans le menu du haut (un petit Android qui sort d'une boîte) :



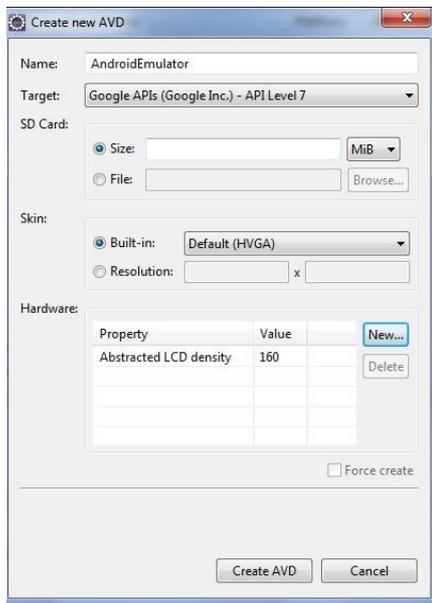
Cliquez sur l'icône en question, une nouvelle fenêtre va apparaître.



Cet écran vous permettra de :

- installer de nouveaux paquets (Available Packages) ;
- mettre à jour vos paquets ;
- voir les paquets déjà installés (Installed Packages) ;
- créer votre émulateur Android et cela grâce à l'onglet « Virtual Devices ». Cliquez sur le bouton « New ».

Une nouvelle fenêtre pour la création de votre émulateur apparaîtra.



- **Name** : le nom de votre émulateur (sans espace).
- **Target** : version du SDK Android de l'émulateur.
- **SD Card (facultatif)** : configuration de la SD Card (Taille, etc.).
- **Skins** : choisissez la taille, résolution de votre émulateur. Des émulateurs préconfigurés se trouvent dans la partie Built-in.
- **Hardware** : cette partie permet de rajouter le matériel spécifique à votre émulateur. Par exemple vous pouvez rajouter un GPS, configurer le clavier, l'accéléromètre, etc.

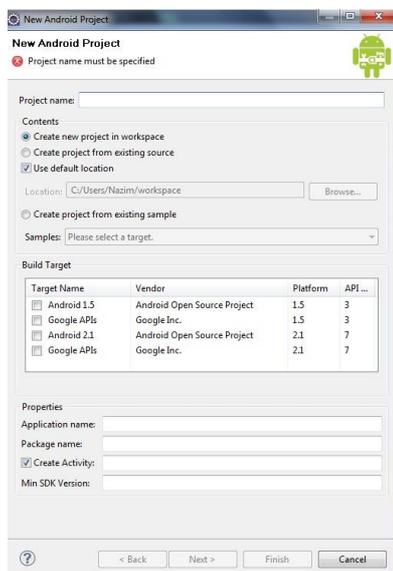
L'émulateur apparaîtra maintenant dans la liste des émulateurs disponibles.

Maintenant on va passer à la partie la plus intéressante.

5. Ma première application sous Android

5.1. Création du projet « Hello World »

Dans Eclipse, cliquez sur « File -> New -> Android Project ». La fenêtre ci-dessous s'affichera :



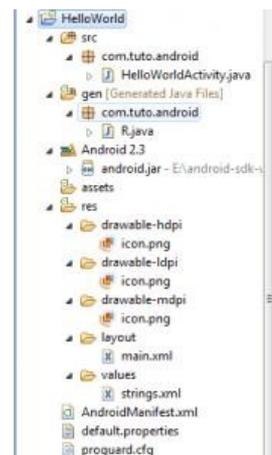
Remplissez les champs :

- **Project name** : le nom du projet. Pour notre exemple on choisira **Hello World**.
- **Build Target** : cochez la SDK que vous souhaitez. On prendra **Android 2.3**.
- **Properties** :
- -----**Application name** : le nom de l'application. On choisira Hello World ;
- -----**Package name** : le nom du package principal de l'application. Il faut que ce dernier comporte au moins deux identifiants séparés par des points. On prendra com.tuto.android ;
- -----**Create Activity** : si vous laissez coché, vous devez spécifier un nom pour l'activité de base de votre application. Nous choisirons HelloWorldActivity ;
- -----**Min SDK Version** : vous pouvez spécifier quelle version minimum du SDK est nécessaire pour le fonctionnement de votre application. Ce champ est facultatif.

Puis cliquez sur « Finish », le projet « Hello World » va apparaître dans l'arborescence d'Eclipse.

5.2. Explication de l'arborescence du projet

Voici le résultat de la création de votre projet et l'arborescence de ce dernier



- **src** : ce dossier contient les sources de votre application (code JAVA) et les packages.
- **com.tuto.android** : un package de votre application. Bien sûr, vous pouvez avoir plusieurs packages dans votre application.
- **HelloWorldActivity.java** : notre principale activité. (je vous conseille d'avoir plusieurs activités pour les différentes parties de votre code).
- **gen** : dossier qui contiendra le fichier R.java (ce fichier est généré automatiquement à partir de vos vues et fichiers de ressource).
- **R.java** : ce fichier est automatiquement généré par le SDK Android à chaque précompilation.
- **assets** : contient des données qui pourront être utilisées dans votre application (images, vidéos, licence, etc.).
- **res** : c'est le dossier qui contiendra les ressources de votre application (images, vidéos, styles).
- **drawable-hpdi** : contient toutes les images,

bitmaps dont vous avez besoin pour votre application en haute résolution.

- **drawable-ldpi** : contient toutes les images, bitmaps dont vous avez besoin pour votre application en basse résolution.
- **drawable-mdpi** : contient toutes les images, bitmaps dont vous avez besoin pour votre application en moyenne résolution.
- **Icon.png** : l'icône de votre application, cette icône sera affichée sur le bureau.
- **layout** : le SDK Android offre une technique de création d'interfaces graphiques à l'aide de fichiers XML. C'est dans ce dossier que vous incluez l'ensemble des fichiers décrivant vos interfaces. Vous pouvez créer d'autres dossiers pour les menus par exemple.
- **Main.xml** : le fichier principal de votre interface.
- **values** : ce dossier contient un ensemble de fichiers décrivant les valeurs (pseudovariables) utilisées par votre application. On peut, par exemple, y mettre des chaînes de caractères (strings.xml), des tableaux (arrays.xml), des entiers, des couleurs, etc.
- **Strings.xml** : fichier qui contient vos déclarations de chaînes de caractères.
- **AndroidManifest.xml** : définit le comportement de votre application au système Android. Ce fichier définit par exemple le nom, l'icône, la version min du SDK, les activités, les services, etc.

5.3. Hello, World!

Le projet exemple créé de base par Eclipse représente un « **Hello World!** ». Vous pouvez le lancer en tant qu'une application Android. Pour cela, il vous suffit de cliquer droit sur le projet, puis sélectionner l'option « Run As -> Android Application » et la l'émulateur devrait se lancer. L'émulateur prendra un peu de temps à se lancer la première fois (ne le fermez pas entre vos différentes modifications).

Voilà le deuxième écran qui devrait s'afficher si tout se passe bien (le premier est pareil mais avec juste écrit « Android »)



Notre Hello Word est bien fonctionnel mais reste encore à le comprendre. Allons voir le code pour comprendre ce qui se passe.

• AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
  xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.tuto.android"
  android:versionCode="1"
  android:versionName="1.0">
  <application android:icon="@drawable/icon"
    android:label="@string/app_name">
    <activity android:name=".HelloWorldActivity"
      android:label="@string/app_name">
      <intent-filter>
        <action android:name="
          android.intent.action.MAIN" />
        <category android:name="
          android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>
</manifest>
```

- La balise « **manifest** » à la ligne deux contient plusieurs arguments, le plus important est « package », qui donne le nom du package dans lequel se trouve votre activité principale.
- La balise « **application** » sert à la déclaration de différentes propriétés de votre application :
- -----**android:icon** : l'emplacement où se trouve l'icône de votre application ;
- -----**android:label** : le nom de votre application (il se trouve dans strings.xml).
- La balise « **activity** » permet de déclarer une activité, à chaque nouvelle activité il faut remettre cette balise.
- -----**android:name** : le nom de la classe Java qui représente l'activité. Le nom doit commencer par un . et on ne met pas le .java à la fin.
- -----**android:label** : le label de l'activité en question.
- -----**intent-filter** : c'est pour spécifier une action.
- -----la sous-balise action est pour spécifier l'action à exécuter, dans notre cas c'est le main.
- -----la sous-balise category est là pour spécifier la catégorie de l'action.
- -----Voici un lien qui explique les différents types d'actions et de catégories : [Lien81](#).
- **strings.xml**

```
<xml version="1.0" encoding="utf-8">
<resources>
<string name="hello">
Hello World, HelloWorldActivity!
</string>
<string name="app_name">Hello World</string>
</resources>
```

- dans les balises **resources**, on met une balise string à chaque fois que l'on a besoin de déclarer une chaîne de caractères ;
- on déclare deux chaînes :
- -----la chaîne **hello** qui contiendra « Hello

World, HelloAndroActivity! » qui est le message qui sera affiché dans l'application ;

- -----la chaîne **app_name** qui contient « Hello Andro » qui représente le nom de l'application.
- **main.xml**
- ----- On dispose de deux modes de visualisation.
- -----**Onglet Layout** : mode visualisation et édition d'interface ;
- -----**Onglet main.xml** : mode code source.
- -----On commence par une balise qui définit le layout : ici **LinearLayout**.
- -----Voici un lien pour la liste des différents layouts : [Lien 82](#).
- -----On déclare une composante **TextView** pour afficher du texte et on
- lui dit qu'elle doit afficher le contenu de **@string/hello**, donc de la
- variable hello qui est déclarée dans **strings.xml** c'est-à-dire « **Hello World, HelloWorldActivity!** ».
- **HelloAndroActivity.java**

```
package com.mti.android;

import android.app.Activity;
import android.os.Bundle;

public class HelloAndroActivity extends Activity
{
    /** Called when the activity is first
    created. */
    @Override
    public void onCreate(Bundle
    savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

- Notre main activité. Elle doit hériter de la classe Activity ou d'une sous-classe de cette dernière.
- [Lien 83](#).
- La méthode « **OnCreate** » est équivalente au main, elle est appelée à la création de votre vue.
- On appelle simplement le **OnCreate** de la classe mère puis on initialise la vue. Puis, on met dedans **R.layout.main**, c'est-à-dire la vue déclarée dans le fichier **main.xml**.
- À chaque fois que vous voyez « **R** », c'est-à-dire

que l'on utilise du code qui a été généré par les différents fichiers xml de ressources.

- « **R.layout** » : on va chercher la vue déclarée dans le dossier layout et qui s'appelle **main** donc notre **main.xml**.
- R.java

```
/* AUTO-GENERATED FILE. DO NOT MODIFY.
 *
 * This class was automatically generated by the
 * aapt tool from the resource data it found. It
 * should not be modified by hand.
 */

package com.mti.android;

public final class R {
    public static final class attr {
    }
    public static final class drawable {
        public static final int icon=0x7f020000;
    }
    public static final class layout {
        public static final int main=0x7f030000;
    }
    public static final class string {
        public static final int
        app_name=0x7f040001;
        public static final int hello=0x7f040000;
    }
}
```

Vous ne devez pas toucher à ce fichier, il est généré automatiquement à partir de vos fichiers qui se trouvent dans le dossier des ressources (res).

- Vous remarquez que toutes les variables déclarées dans strings.xml sont présentes, que l'interface déclarée dans main.xml aussi. Ce qui explique l'utilisation de la ligne R.layout.main dans le HelloWorldActivity.java ainsi que l'icône de l'application.

6. Conclusion

Voilà on s'arrête ici pour ce premier tutoriel, d'autres tutoriels vont suivre très prochainement et aborderont des sujets plus approfondis.

Retrouvez l'article de Nazim Benbourahla en ligne : [Lien 84](#)

Installation et utilisation d'eclipse pour le développement en Java

La version originale de cet article peut être trouvée ici : [Lien 85](#). Copyright Henri Garreta ([Lien 86](#)), Université de la Méditerranée (Aix-Marseille Université) ([Lien 87](#)) et Faculté des Sciences de Luminy ([Lien 88](#)).

La documentation en ligne d'eclipse est abondante mais son approche n'est pas immédiate. Sans souci d'exhaustivité, nous donnons ici quelques indications pour débiter rapidement dans l'utilisation de cet excellent outil de développement.

À la date du 16 juillet 2009, les versions courantes des logiciels en question sont :

Java JDK 1.6.0 (ou 6) update 14 ;
Eclipse platform 3.5.0 « Galileo ».

Les versions décrites ici ne sont peut-être pas exactement celles-là mais elles leur sont fonctionnellement équivalentes.

Note de la rédaction

Cet article datant de quelques années, les versions des logiciels ont évolué, mais l'ensemble de cet article est réalisable avec les versions actuelles.

1. Se procurer le kit de développement Java

Eclipse ne contient ni le compilateur Java ni les autres outils basiques. Pour développer des programmes en Java, il faut donc installer au préalable un kit de développement. Nous conseillons celui de Sun Microsystems, la maison mère de Java, qui est complet, à jour (par définition) et gratuit.

Vous pouvez l'obtenir chez Sun : site java.sun.com ([Lien 89](#)), menu Downloads, rubrique Java SE (comme Standard Edition). Le produit à télécharger s'appelle, lors de la publication de cette notice, JDK 6 update 14.

Attention, ne confondez pas le JDK (Java Development Kit) avec le JRE (Java Runtime Environment), appelé parfois « plugin Java », qui ne contient que le nécessaire pour exécuter les programmes Java. Ne vous occupez pas de télécharger le JRE, à l'intérieur du JDK il y en a un exemplaire.

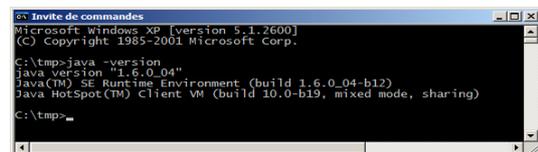
Le fichier qui vous concerne se nomme :

- dans le cas de Windows : **jdk-6u14-windows-i586.exe** (73,5 Mo) ;
- dans le cas de Linux : **jdk-6u14-linux-i586.bin** (77 Mo).

Il s'agit dans les deux cas d'un installateur auto-extractible : après le téléchargement, il suffit de le lancer et de suivre les instructions qui s'affichent. Au besoin, des renseignements supplémentaires sur l'installation du JDK

sont donnés sur le site de Sun, aussi bien pour Windows ([Lien 90](#)) que pour Linux ([Lien 91](#)).

Une fois l'installation terminée, vérifiez sa réussite en tapant « **java -version** » dans une console de commandes. Vous devez obtenir un message vous annonçant le numéro de version de la machine Java mise en place. Dans le cas de Windows, cela ressemblera à ceci :



```
Microsoft Windows [version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\tmp>java -version
java version "1.6.0_04"
Java(TM) SE Runtime Environment (build 1.6.0_04-b12)
Java HotSpot(TM) Client VM (build 10.0-b19, mixed mode, sharing)

C:\tmp>
```

Pour développer des programmes en Java, il vous faut disposer également de la documentation de l'API (*Application Programmer Interface*, c'est-à-dire le volumineux ensemble de paquetages, classes, méthodes et variables qui constituent la bibliothèque système). Vous pouvez la consulter en ligne ([Lien 92](#)) ou bien la télécharger depuis le site java.sun.com ([Lien 89](#)), menu **Downloads**, rubrique **Java SE**, produit **Java SE 6 Documentation** (quel que soit votre système d'exploitation, le fichier s'appelle **jdk-6u10-docs.zip** et pèse 56 Mo).

Note (cas de *Windows*). Si vous souhaitez pouvoir employer le compilateur et les autres outils Java en dehors d'*eclipse*, c'est-à-dire en tapant des commandes dans une console *Invite de commandes*, alors vous devez procéder à la manipulation supplémentaire suivante : repérer le répertoire ® d'installation de Java et ajouter le chemin ® **\bin** dans la définition de la variable *Path*.

Si vous avez laissé l'installateur de Java faire à sa guise, ® doit être quelque chose comme **C:\Program Files\Java\jdk1.6.0_04**.

Vous pouvez examiner et modifier la valeur de la variable *Path* en cliquant avec le bouton droit sur l'icône du *Poste de travail*, puis *Propriétés* > *Avancé* > *Variables d'environnement* > *Variables système* ; sélectionner la ligne *Path* puis faire *Modifier*.

Dans le cas de *Linux*, une manipulation analogue est nécessaire après l'installation du *JDK*. Nous ne l'expliquons pas car elle fait partie des opérations courantes sur ce système.

2. Télécharger eclipse

Eclipse est un logiciel libre que vous pouvez télécharger depuis le site www.eclipse.org ([Lien 93](#)), onglet *Downloads*. Le produit qui nous intéresse est *Eclipse IDE for Java Developers* (92 MB).

Le fichier téléchargé se nomme :

- dans le cas de Windows : **eclipse-java-galileo-win32.zip** ;
- dans le cas de Linux : **eclipse-java-galileo-linux-gtk.tar.gz** ;
- dans le cas de Max OS X : **eclipse-java-galileo-macosx-carbon.tar.gz**.

Nous ne vous conseillons pas de télécharger une version française d'*eclipse*. Il existe bien des plugin de francisation de l'interface, mais outre le fait qu'elles sont assez imparfaites, elles servent surtout à vous empêcher d'utiliser la dernière version du logiciel.

3. Installer eclipse

A partir d'ici, les explications sont communes aux divers systèmes d'exploitation, ou bien ne concernent que Windows XP et Vista.

Pour installer *eclipse*, il suffit de décompresser l'archive *zip* ou *tar.gz* téléchargée. Cela crée un dossier, nommé **eclipse**, que nous vous conseillons de placer aussi haut que vous le pouvez dans la hiérarchie de fichiers de votre système.

Dans la suite de cette note nous supposons que vous avez fait ainsi et que vous avez donc un dossier nommé **C:\eclipse**.

Pour faciliter le lancement d'*eclipse*, créez un raccourci vers le fichier **C:\eclipse\eclipse.exe** et placez-le sur le bureau, dans le menu démarrer ou ailleurs, selon vos goûts.

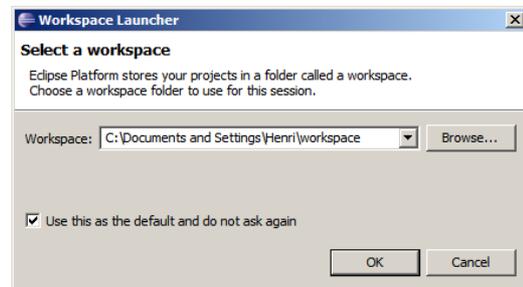
L'installation d'*eclipse* est donc bien plus légère que celle de beaucoup de logiciels ; en particulier, sous Windows, elle ne produit pas d'inscription dans la base de registres. Par conséquent, pour désinstaller complètement *eclipse* il suffira, le moment venu, de mettre à la corbeille le dossier **C:\eclipse** et les divers espaces de travail (dossiers **workspace**, voir ci-dessous) créés ultérieurement.

4. Premier lancement d'eclipse

Lancez *eclipse*, par exemple en double-cliquant sur le raccourci que vous venez de créer. Au bout de quelques instants, on vous demandera de situer l'espace de travail dans lequel seront vos fichiers. Si vous travaillez sur un

ordinateur partagé, il est conseillé de mettre l'espace de travail dans votre dossier Documents (sur Windows XP il se trouve dans le dossier Documents and Settings). Si vous êtes le seul utilisateur de votre système, mettez l'espace de travail où bon vous semble.

Sauf indication contraire, les fichiers sources de vos programmes se trouveront dans l'espace de travail. Il est donc important de se souvenir de l'emplacement de ce dernier pour accéder aux sources (par exemple, pour les transporter, les copier, etc.).

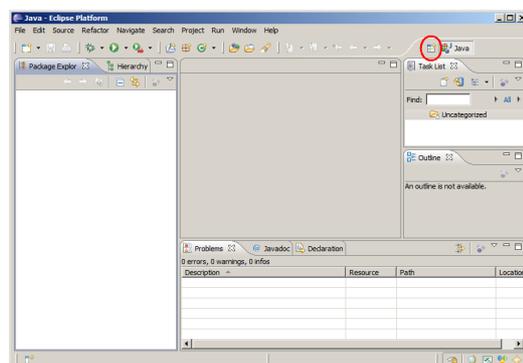


Si vous cochez la case « *Use this as the default and do not ask again* » *eclipse* ne vous posera plus cette question (mais il y a toujours un moyen pour changer ultérieurement l'espace de travail : *File* > *Switch Workspace* > *Other...*).

Au bout de quelques instants (la première fois ce n'est pas très rapide) vous obtenez un écran qui présente le produit, comme ceci :



Vous pouvez feuilleter cette présentation, elle est faite pour cela. Quand vous en aurez assez, cliquez sur le lien *Workbench* (la flèche représentée à droite de l'écran). Le contenu de la fenêtre devient tout de suite beaucoup plus sérieux :

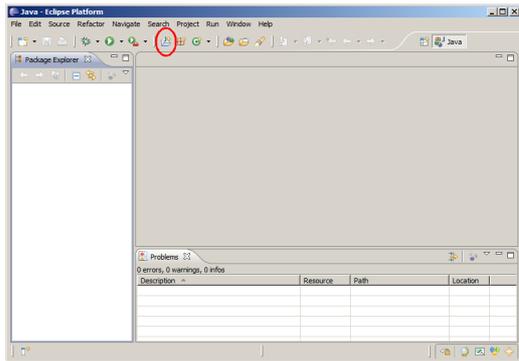


5. Configurer eclipse pour faire du Java

Eclipse est un environnement qui permet une grande variété d'activités de développement (pour vous en

convaincre, faites un tour chez eclipse plugin central ([Lien 94](#)), chez eclipse plugins ([Lien 95](#)) ou bien sur le site francophone eclipsetotale.com ([Lien 96](#)). En standard, *eclipse* est prêt pour le développement en Java, encore faut-il veiller à ce que la *perspective* (c'est-à-dire l'arrangement des vues montrées à l'écran) soit celle qui convient le mieux à Java. Si ce n'est pas le cas, agissez sur la petite icône en haut à droite cerclée de rouge sur la figure 4, étiquetée *Open perspective* et choisissez *Java*.

Fermez les vues *Task List* et *Outline* (à droite) ; pour afficher la structure des classes, la vue *Package explorer* (à gauche) suffit. Vous obtenez un cadre de travail tout à fait commode pour développer en *Java* :



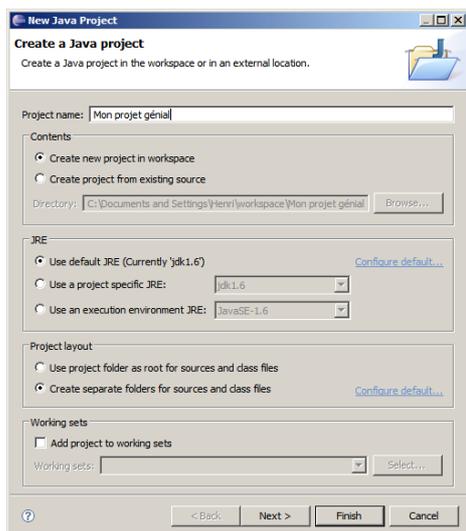
6. Développer un programme Java

Pour commencer, créer un projet.

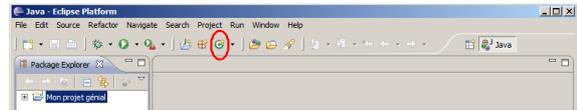
Note pour les étudiants. Ne vous sentez pas obligés de créer un nouveau projet chaque fois que vous commencez un nouvel exercice de programmation : vous pouvez très bien avoir un seul projet, contenant tous les exercices que vous faites dans le cadre d'un enseignement. D'autant plus que cela ne vous empêchera pas de bien ranger vos fichiers : un projet peut contenir plusieurs packages java (qui se traduiront dans le système de fichiers par des répertoires différents).

Pour créer un projet, cliquez sur le premier des boutons d'assistants Java (cerclé de rouge sur la figure 5) étiqueté « *New Java Project* ».

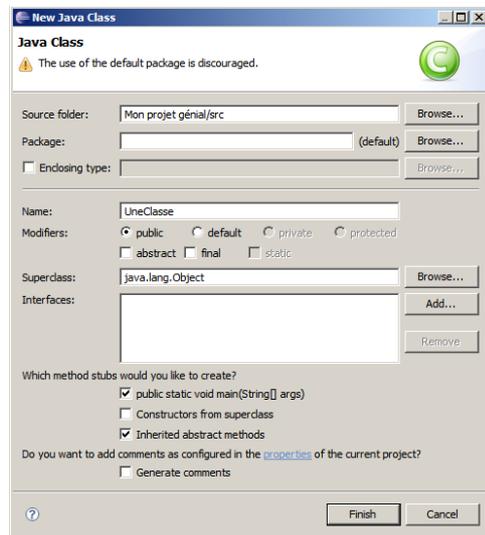
Vous obtenez le panneau *New Java Project* où, au minimum, vous devez donner un nom pour votre projet :



La méthode rapide consiste à donner un nom de projet (si possible, moins bête que *Mon projet génial...*) et cliquer sur le bouton *Finish*. Notez que les autres « questions » posées dans ce panneau sont intéressantes. La troisième, notamment, permet de conserver *séparément* les fichiers sources (précieux) et les fichiers classes (qu'en cas de perte on peut toujours refaire).

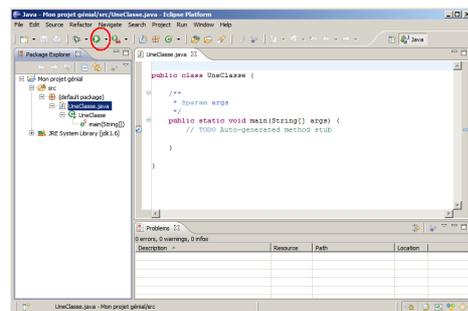


Dans un projet sérieux nous commencerions par créer des packages (deuxième bouton des assistants Java, « *New Java Package* »). Mais, puisque nous débutons, allons à l'essentiel et ajoutons directement une ou plusieurs classes au projet : c'est le troisième des boutons d'assistants Java, « *New Java Class* » (cerclé de rouge dans la figure 7), qui fait cela. La méthode rapide consiste à donner le nom de la classe et cocher la case étiquetée *public static void main(String[] args)* :



Notez qu'*eclipse* critique notre démarche, nous indiquant que *l'emploi du package par défaut (sans nom) est découragé*. Cela ne fait rien, nous construisons ici une application de débutant.

Eclipse crée alors un fichier source contenant une classe rudimentaire, correcte mais creuse, que vous n'avez plus qu'à compléter pour en faire le programme voulu :



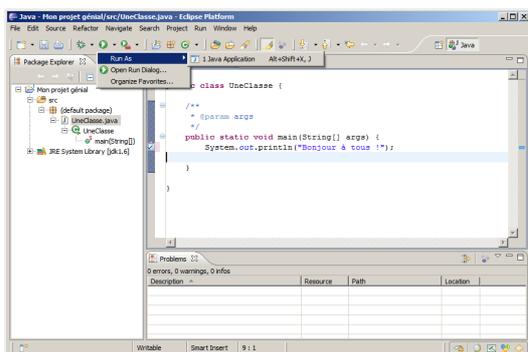
Note. Lorsqu'un commentaire contient l'expression *TODO*, *eclipse* affiche une marque bleue dans la marge qui permet de se rendre rapidement à cet endroit. C'est très pratique

pour retrouver dans les gros fichiers ces commentaires qui signalent des morceaux en chantier. Pour essayer votre programme vous allez taper le classique **System.out.println("Bonjour à tous !");** à l'intérieur de la fonction **main**. Au fur et à mesure que vous tapez, remarquez comment :

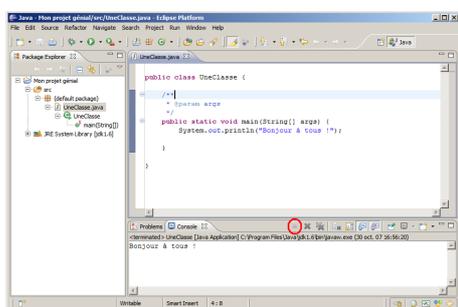
la vue *Package Explorer* montre les packages (répertoires) qui composent votre projet, les classes que ces paquetages contiennent, les membres de ces classes, etc. Bien entendu, double cliquer sur une de ces entités vous positionne dessus dans le texte source ;

- si vous marquez une pause lorsque vous tapez un point, *eclipse* vous montre la liste de ce que vous pouvez taper ensuite ;
- si vous laissez traîner le curseur sur un identificateur, *eclipse* affiche la documentation correspondante ;
- si vous faites une faute, *eclipse* vous la signale immédiatement et, dans le cas d'erreurs sémantiques, vous suggère des corrections ;
- le simple fait de sauver le programme en provoque la compilation.

Pour exécuter le programme, assurez-vous que la vue éditeur contient une classe exécutable (c'est-à-dire une classe *publique* avec une méthode **public static void main(String[] args);**) et alors activez la commande *Run as > Java Application* du menu attaché au bouton cerclé de rouge sur la figure 9 :



L'application s'exécute et, si des sorties sont à afficher, une vue Console apparaît au-dessous de la vue éditeur :



Notez que dans la vue *Console* il y a un bouton, cerclé de rouge sur la figure 11, qui permet d'arrêter une application qui bouclerait indéfiniment. Ce bouton est rouge quand l'application est vivante, gris (estompé) lorsque l'application est morte.

7. Où sont mes fichiers sources ?

Cette question se pose par exemple lorsque, après avoir

développé une application dans *eclipse*, vous souhaitez récupérer vos fichiers sources pour les amener sur un autre système, les compiler dans un autre environnement ou tout simplement les ranger dans vos archives.

La réponse se trouve dans les figures 2 et 6 : si lors de la création du projet vous avez laissé l'option par défaut « *Create new project in workspace* » (cf. figure 6) alors les fichiers sources, rangés dans des dossiers correspondant aux packages, sont dans le dossier *workspace*, lui-même placé à l'endroit que vous avez indiqué au lancement d'*eclipse* (cf. figure 2).

8. Comment amener dans eclipse des fichiers créés ailleurs ?

Deux cas possibles : ces fichiers forment déjà un projet *eclipse* (par exemple créé sur un autre système), ou bien il ne s'agit que d'un ensemble de fichiers sources en vrac.

8.1. Vous avez déjà un projet eclipse

Copiez le dossier du projet *eclipse* où vous voulez (par exemple dans le dossier *workspace*, mais ce n'est pas une obligation), puis faites la commande *File > Import...* Ensuite, choisissez *General* puis *Existing Projects into Workspace*. Le projet que vous venez d'importer apparaît dans la fenêtre *Package explorer*, c'est terminé.

8.2. Vous n'avez qu'un ensemble de fichiers sources Java

Prenez un projet qui existe déjà, ou bien créez un nouveau projet. Ensuite :

- soit, à l'aide de la commande *File > Import... > General > File system*, vous naviguez à la recherche du(des) fichier(s) en question et vous les importez dans ce projet ;
- soit, plus simplement : vous copiez les fichiers dans le dossier où sont les sources d'un des projets connus dans *eclipse*, puis vous sélectionnez ce projet dans la vue *Package Explorer* et enfin vous exécutez la commande *File > Refresh*.

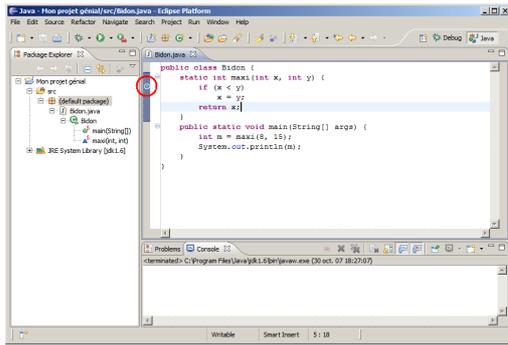
9. Débugger les programmes

Un programme « bogué » est un programme qui ne donne pas les résultats qu'il devrait. « Débugger » un programme c'est chercher les erreurs de programmation à l'origine de tels dysfonctionnements. Pour aider le programmeur dans cette recherche, *eclipse* offre un mode debug permettant, entre autres choses :

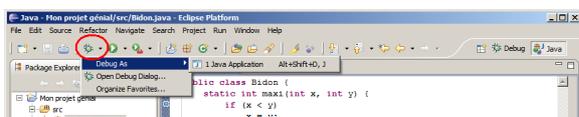
- la pose de marques, appelées *points d'arrêt*, sur des lignes du programme source, de telle manière que l'exécution s'arrêtera lorsque ces instructions seront atteintes ;
- lors de tels arrêts, l'examen des valeurs qu'ont alors les variables locales et les membres des objets ;
- à partir de là, l'exécution du programme pas à pas (c'est-à-dire ligne à ligne).

Pour déboguer simplement un programme, il suffit de poser un point d'arrêt au début de l'endroit qu'on souhaite examiner en détail. Pour cela, il faut double-cliquer dans la

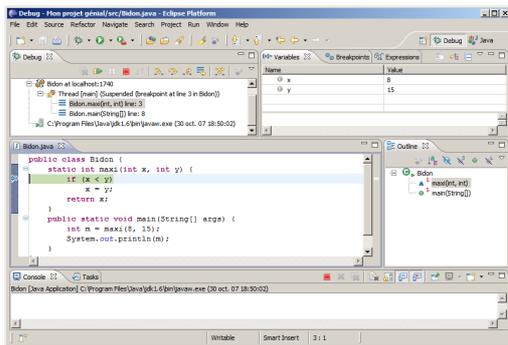
marge, à gauche de la ligne en question, ce qui fait apparaître un disque bleu (cerclé de rouge dans la figure 12) qui représente le point d'arrêt.



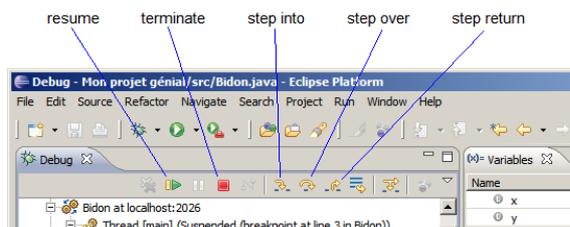
Il faut ensuite lancer le débogage, à l'aide du bouton à gauche de celui qui lance l'exécution, représentant une punaise (bug) :



L'exécution est alors lancée et se déroule normalement jusqu'à atteindre le point d'arrêt. Eclipse demande alors la permission de changer de *perspective* (ensemble et disposition des vues montrées) et adopte l'apparence de la figure 14 :



La vue *Debug*, en haut à gauche de la fenêtre, montre la *pile d'exécution*, c'est-à-dire, pour chaque thread, l'empilement des méthodes qui se sont mutuellement appelées (méthodes commencées et non terminées). Dans la figure 14, par exemple, on attire notre attention sur la méthode *Bidon.main*, plus précisément la ligne 8 du fichier source, où a été appelée la méthode *Bidon.maxi*, dans laquelle l'exécution est arrêtée, à la ligne 3.



En haut de cette vue (figure 15) se trouvent des boutons très utiles. Parmi les principaux :

- *step over* : faire avancer l'exécution d'une ligne. Si cette dernière contient un appel de méthode, *ne pas détailler* l'activation de celle-ci, c'est-à-dire considérer l'appel comme une instruction indivisible ;
- *step into* : avancer l'exécution d'une ligne. Si un appel de méthode est concerné, détailler son activation, c'est-à-dire aller dans la méthode et s'arrêter sur sa première ligne ;
- *step return* : relancer l'exécution normale, jusqu'à la fin de la méthode dans laquelle on est arrêté et le retour à la méthode qui a appelé celle-ci ;
- *resume* : relancer l'exécution normale, jusqu'à la fin du programme ou le prochain point d'arrêt ;
- *terminate* : terminer l'exécution.

En haut et à droite de la fenêtre principale se trouvent les vues *Variables* et *Expressions*. La première affiche les valeurs courantes des variables locales de la méthode en cours, la deuxième affiche les valeurs courantes des expressions sélectionnées avec la commande *Watch* (cliquer avec le bouton droit sur l'expression à surveiller).

Le débogueur d'*eclipse* possède bien d'autres commandes très puissantes, comme les *points d'arrêt conditionnels* et la possibilité de *modifier* les valeurs des variables du programme. Prenez un peu de temps pour les explorer, c'est payant.

10. Le « refactoring »

Réviser (refactor) un programme correct c'est modifier son texte source sans changer son fonctionnement et ses résultats. Par exemple, changer le nom d'une variable ou d'une méthode parce que, par suite de l'évolution du programme (ou du programmeur), le nom initialement choisi est devenu moins adapté ou expressif qu'un autre.

Le *refactoring* se traduit généralement par des opérations globales, fastidieuses, qu'il n'est pas facile d'automatiser. Heureusement *eclipse* offre de puissantes fonctions pour effectuer ce travail. Voici leur description, directement traduite de l'aide en ligne du logiciel.

Commandes du menu Refactor

Voir tableau en ligne : [Lien 97](#)

Commentez cette news de Henri Garreta en ligne : [Lien 98](#)

Comment configurer Outlook pour qu'il agisse dès l'arrivée de messages dans votre boîte aux lettres

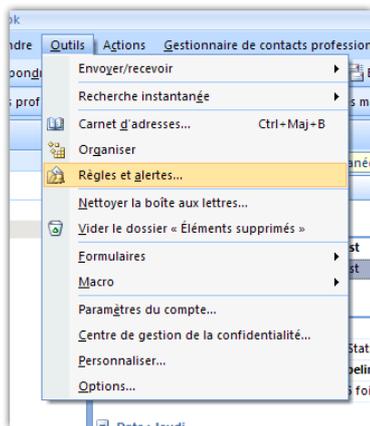
Dans cet article, nous allons parcourir la méthode de création de règles pour Microsoft Outlook.

1. Introduction

Régulièrement, sur le forum Outlook apparaissent des questions concernant les règles de réception de messages. Pour appliquer certaines actions lors de l'arrivée de messages, il existe deux possibilités, la première, c'est d'utiliser le VBA et écrire un morceau de code qui effectuera certaines actions à votre place. La seconde est d'utiliser l'assistant " Règles et Alertes " .

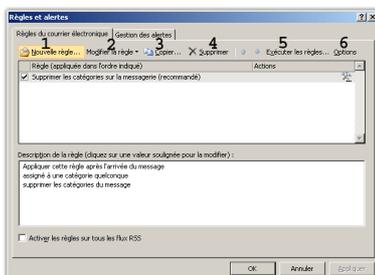
2. L'assistant " Règles et Alertes " .

Dans le menu Outils, il y a une entrée " Règles et Alertes " , c'est cette entrée qui va nous amener à l'assistant. (Sous Outlook 2002, on y accède par : "Assistant Gestion des messages".)



Menu " Règles et Alertes "

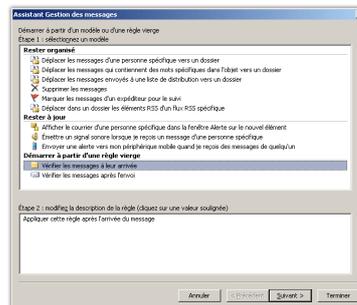
La boîte de dialogue qui s'ouvre ensuite offre plusieurs possibilités d'actions pour les règles.



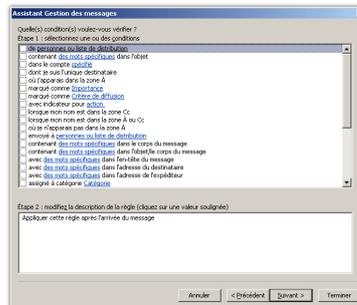
1. Création d'une nouvelle règle
2. Modification d'une règle existante
3. Copie d'une règle
4. Suppression d'une règle
5. Exécution des règles
6. Options

Si vous utilisez un serveur Exchange, certaines règles fonctionnent sur le serveur, même durant votre absence, alors que d'autres nécessitent que Outlook soit ouvert.

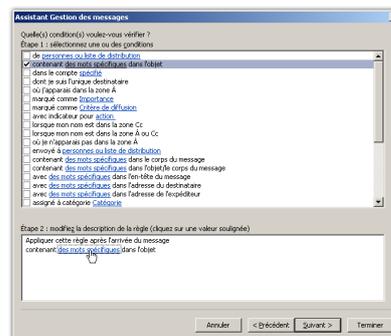
Que vous optiez pour une nouvelle règle ou pour la modification d'une règle, le principe est toujours le même. Pour ce qui va suivre, je pars d'une nouvelle règle.



J'ai choisi d'exécuter la règle lors de la réception des messages, on peut également l'exécuter lors de l'envoi.

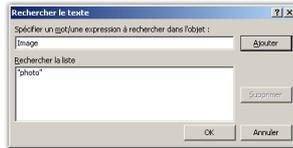


Vous pouvez cocher les critères qui vous conviennent le mieux. Ces critères vont permettre d'agir comme un filtre sur les messages entrants. Si vous ne cochez aucune ligne, cette règle s'appliquera à tous les messages.

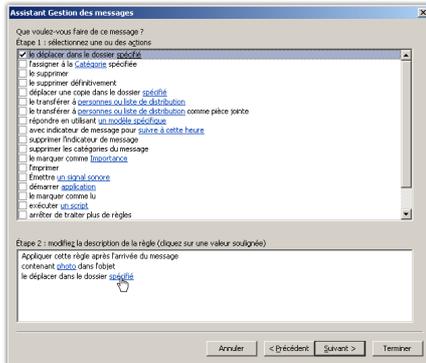


Les paramètres des filtres sont modifiables dans la fenêtre inférieure, le pointeur se transforme en main, et ouvre une

boîte de dialogue supplémentaire.



L'écran suivant vous demande ce que vous voulez faire.



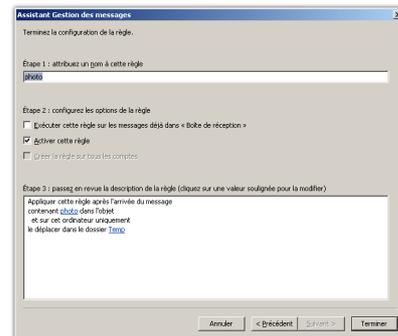
Si vous désirez copier le message dans un dossier spécifique, une nouvelle boîte de dialogue s'affiche.



Vous pouvez comme pour toutes les règles définir des exceptions.



Et terminer par le nom de votre règle.



Lors de cette dernière étape, vous avez la possibilité d'exécuter la règle immédiatement, sur les messages déjà présents, ou d'activer la règle pour les prochaines réceptions.

Retrouvez l'article d'Olivier Lebeau en ligne : [Lien 99](#)

Fusion et Publipostage, par l'interface et en VBA

Word peut se révéler un outil de choix pour la diffusion en masse de documents personnalisés. Écrire une seule lettre qui sera différente pour chaque destinataire.

1. Introduction

Suite à de nombreuses questions liées au publipostage, j'ai décidé de me lancer dans un tutoriel. Je vais tenter d'apporter quelques éclaircissements sur les méthodes à utiliser. Certaines sont accessibles aux débutants, d'autres font appel à des techniques avancées de VBA.

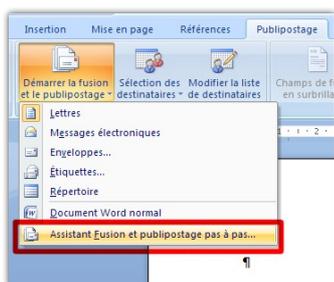
Toutes les illustrations sont issues de Word 2007, mais les méthodes utilisées sont applicables aux versions antérieures.

Le publipostage est une technique qui permet de créer un grand nombre de lettres types dont le contenu est personnalisé. Les informations utilisées pour le publipostage ne se limitent pas aux données personnelles, elles peuvent être d'une tout autre nature : factures, contrats...

Dans les parties qui vont suivre, le document de fusion principal sera toujours un document Word et l'application de départ sera également Word, seules les sources de données vont changer.

1.1. Les commandes

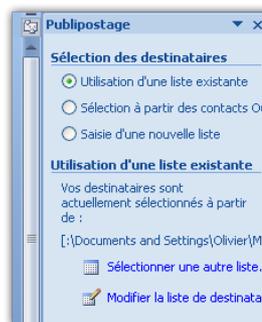
Dans l'onglet Publipostage, Word possède une commande qui permet d'activer un assistant qui vous guidera pas à pas.



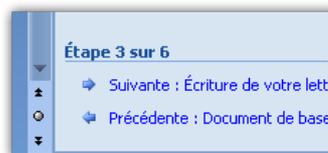
Dans la version 2007, cet assistant se présente sous la forme d'un volet.



Ce volet comporte deux parties importantes. La partie supérieure contient l'étape en cours.



La partie inférieure permet de naviguer d'une étape à l'autre.

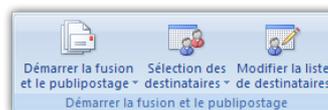


Le ruban contient tous les outils nécessaires pour effectuer le publipostage.



Le groupe Créer

Ce groupe permet de choisir entre enveloppes et étiquettes.

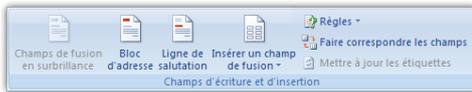


Le groupe Démarrer la fusion et le publipostage

Le premier bouton du groupe permet de choisir le type de document que vous allez créer. Il ouvre une liste déroulante permettant de faire un choix. C'est également ce bouton qui donne accès à l'assistant.

Le second bouton permet la sélection des destinataires ou l'introduction d'une nouvelle liste.

Enfin, le dernier bouton permet de travailler la liste en ajoutant un tri.



Le Groupe champs d'écriture et d'insertion

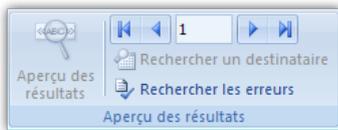
Le premier bouton de ce groupe ne devient actif que si vous avez au moins un champ de fusion inséré sur votre document.

Le second permet d'insérer le bloc d'adresse.

Le troisième bouton offre la possibilité d'insérer une ligne de salutation.

Le dernier permet de choisir l'ajout d'un champ dans la liste. Vous pouvez réutiliser un champ du bloc adresse comme le prénom ou choisir un autre champ contenant d'autres données.

Finalement, il est parfois utile de prévisualiser les résultats avant de lancer l'impression ou l'envoi :

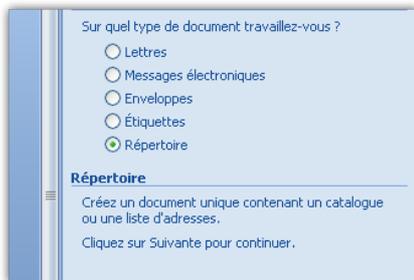


Le groupe Aperçu des résultats

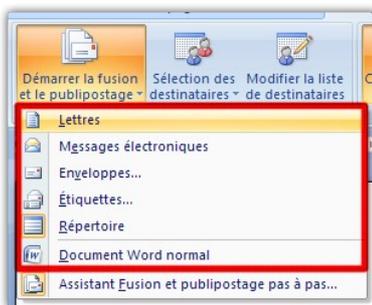
Dans certains cas de figure, la prévisualisation n'est pas correcte, seul le résultat final l'est.

2. Le type de document

Pour le type de document, Word est plein de ressources.



ou encore



Ces choix correspondent à : une lettre, par exemple si votre activité change d'adresse et que vous souhaitez prévenir vos clients, un message électronique comme une "News Letter", parallèlement à l'édition de lettres vous pouvez y associer des enveloppes, faire des planches d'étiquettes ou finalement pour "créer" un répertoire (répertoire téléphonique).

3. Nouveau ou document en cours

Vous pouvez créer un document fondé sur le document en cours d'édition, créer un nouveau document fondé sur un modèle existant ou créer un nouveau document d'après un document existant.



Cette étape ne devrait normalement pas poser de problème.

4. Les sources de données

Nous allons parcourir quelques types de sources de données utilisables avec Word.

Une fois la source de données choisie, si vous enregistrez votre document, vous obtenez un document principal de fusion.

4.1. Word

Pour la source de données, nous pouvons utiliser un document Word. Il est beaucoup plus aisé d'utiliser un tableau pour gérer vos données. Si vous respectez certaines conventions, il sera d'autant plus facile d'utiliser ce document.

Liste préformatée d'un tableau

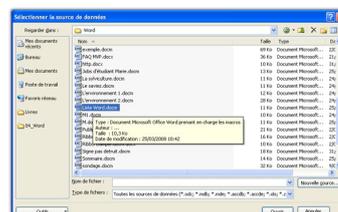
- Nom
- Prénom
- Adresse
- No
- Cp
- Ville
- Tel

Nom	Prénom	Adresse	No	Cp	Ville
Almar	Jean	Rue-en-fer	100	1000	Bruxelles

La liste peut aussi être formatée en CSV (Valeurs Séparées par des Virgules).

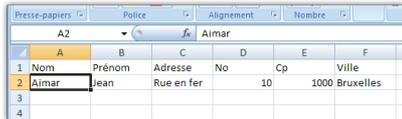
```
Nom,prenom,adresse,no,cp,ville
Almar,Jean,Rue-en-fer,10,1000,Bruxelles
```

Il ne reste qu'à choisir le document Word contenant la liste.



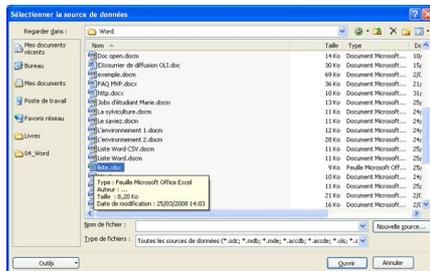
4.2. Excel

Cette liste peut également être une feuille Excel.

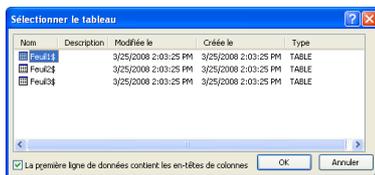


	A	B	C	D	E	F
1	Nom	Prénom	Adresse	No	Cp	Ville
2	Aimar	Jean	Rue en fer	10	1000	Bruxelles
3						
4						

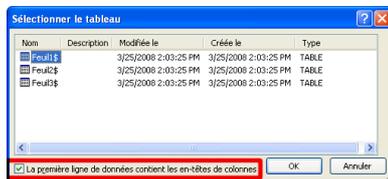
On choisit cette liste de la même manière que pour un fichier Word.



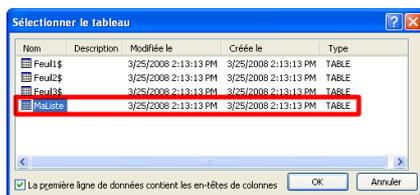
Et votre liste peut se trouver sur n'importe quelle feuille du classeur.



Si votre première ligne n'est pas une ligne de données, vous devez le préciser à l'ouverture.



Il est même possible d'utiliser une plage de données.

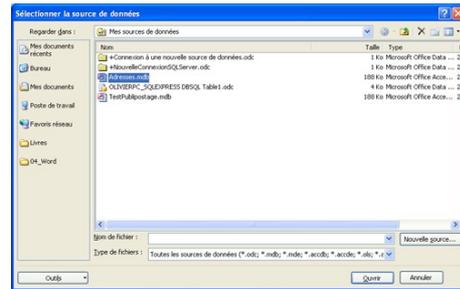


4.3. Access

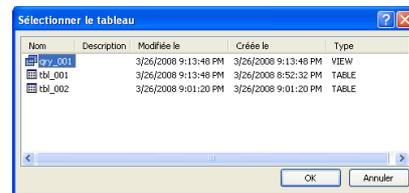
L'utilisation d'une base de données Access pour la fusion et le publipostage est bien évidemment possible. Dans l'onglet publipostage, après un clic le bouton Sélection de destinataires, choisissez Utiliser la liste existante.



Vous obtenez alors une boîte de dialogue permettant de choisir un fichier, le répertoire par défaut est le répertoire contenant les informations sur les sources de données.



Si votre base de données possède plusieurs tables, vous pouvez choisir la table qui contient vos adresses ou données pour le publipostage. Les requêtes sont également proposées.



4.4. Outlook

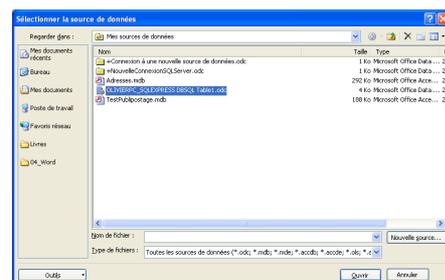
Outlook contenant aussi des adresses, vous pouvez sélectionner le carnet d'adresses Outlook pour un publipostage. Cette méthode se révèle très intéressante si vous désirez faire un E-mailing.



Si vous possédez plusieurs carnets d'adresses, vous aurez le choix.

4.5. SQL Server

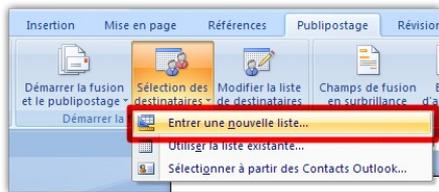
Pour cette partie, nous allons utiliser une source de données située sur une base SQL (SQL Server 2005 Express). J'ai choisi cette version parce qu'elle est disponible pour tous du fait de sa gratuité. Pour lier des données issues d'une base de données SQL Server, on procède de la même manière que pour les autres sources de données.



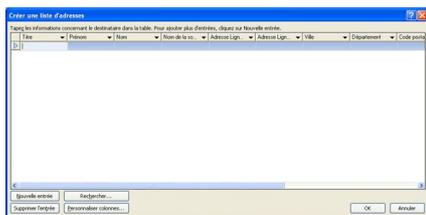
Tous les exemples ont été réalisés à partir de SQL Server Express 2005 : [Lien 100](#).

4.6. Création d'une liste

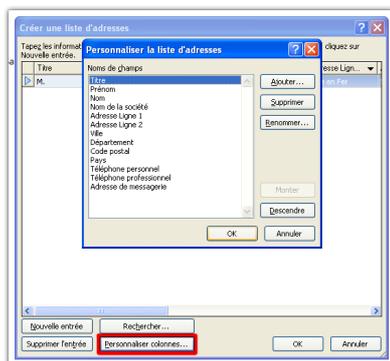
Si vous n'avez pas encore de liste et que vous désirez créer une nouvelle liste, Word possède un outil qui va vous permettre de saisir une liste utilisable dans une fusion ou un publipostage.



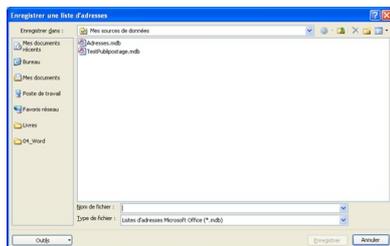
Vous ouvrez un formulaire qui va vous permettre d'entrer des données.



Ce formulaire sommaire contient cependant tous les champs nécessaires pour faire votre publipostage. Si les colonnes proposées ne vous conviennent pas, vous avez la possibilité de les personnaliser.



Dès que votre liste est terminée, le clic sur le bouton OK ouvre une boîte de dialogue vous demandant d'enregistrer vos données. Le seul format proposé est le format *.mdb qui est le format de base de données des versions antérieures d'Access.

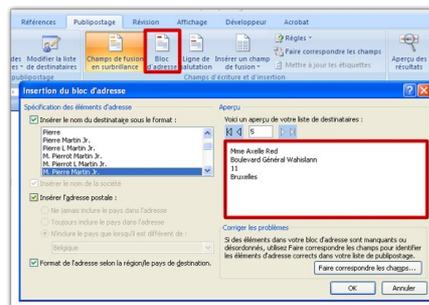


Par défaut, Word contient les outils minimums pour l'utilisation et la création d'une base de données Access.

5. Le document

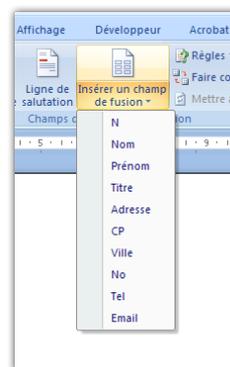
Vous pouvez éditer votre document à n'importe quelle étape du publipostage. Il existe des champs par défaut et des blocs pour vous faciliter la tâche. Le plus intéressant

est le bloc d'adresse.

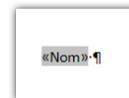


Si vous utilisez le bouton Bloc d'adresses, vous pouvez visualiser le résultat obtenu et parcourir les différents enregistrements qui constituent votre liste.

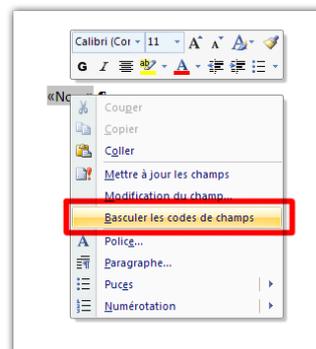
Vous avez également la possibilité d'insérer un champ contenant des données autres que le bloc adresse.



Lorsque vous avez choisi un champ, il s'affiche dans votre document entouré de guillemets.



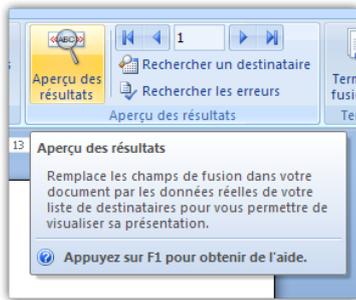
Si vous voulez visualiser le contenu de votre champ, vous devez faire un clic droit sur le champ et choisir "Basculer les codes de champ".



Vous obtenez alors l'affichage du code du champ de fusion et de publipostage.



Après avoir inclus au moins un champ dans votre document, vous pouvez visualiser l'aspect final de votre document. Vous avez à votre disposition un groupe de contrôle pour vous y aider.

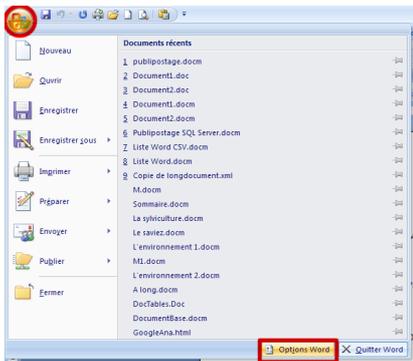


6. L'utilisation avancée des champs

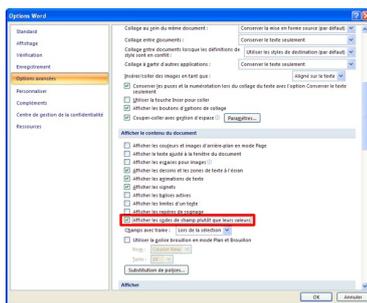
Jusqu'à présent, nous avons utilisé les champs de publipostage de façon très simple. Word permet une utilisation un peu plus avancée des champs.

Pour l'utilisation avancée du publipostage, je vous conseille de modifier les paramètres d'affichage de votre document.

Bouton Office, Options Word



Dans la partie Options Avancées et Afficher le contenu du document, choisissez Afficher les codes de champ plutôt que leurs valeurs.

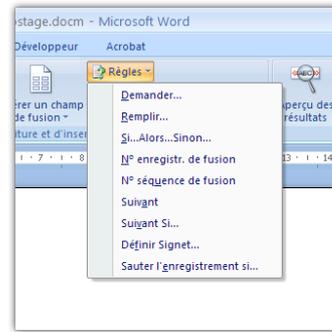


La combinaison de touche **Alt + F9** produit le même résultat.

6.1. Les règles

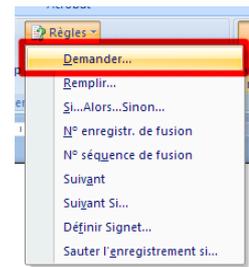
Word offre la possibilité de fixer des règles pour le publipostage.

Toutes les règles ne sont pas compatibles entre elles.



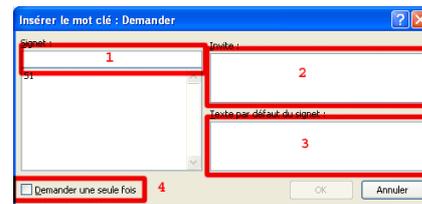
6.1.1. Demander

Dans le cadre d'un publipostage, ce champ n'a pas de fonction immédiate et ne doit pas être utilisé seul. Je vous conseille de l'utiliser avec un champ REF.



Insérer le champ demander

Vous ouvrez alors une boîte de dialogue.



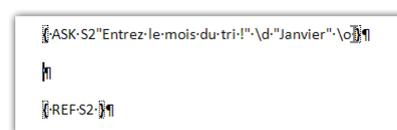
Le remplissage

1. Le nom du signet qui sera inséré.
2. Le titre de la boîte de dialogue (je vous conseille d'y mettre votre question)
3. Le message par défaut (si vous n'entrez rien ce sera ce texte qui sera inséré dans le signet)
4. Si cette option n'est pas cochée, Word vous demandera un texte pour chaque document.



La boîte remplie

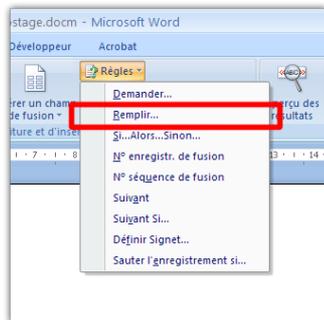
Si vous n'utilisez pas de champ REF, vous ne verrez pas la valeur entrée dans votre document final.



Avec champ REF

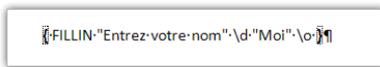
6.1.2. Remplir

En comparaison avec le champ précédent, le champ remplir ne se retrouve pas dans un signet. Il n'est donc pas possible de l'utiliser avec le champ REF. Le texte entré est intégré à votre document. Ce champ pourrait être utilisé pour renseigner le nom du correspondant qui a publié les documents.



Le champ remplir

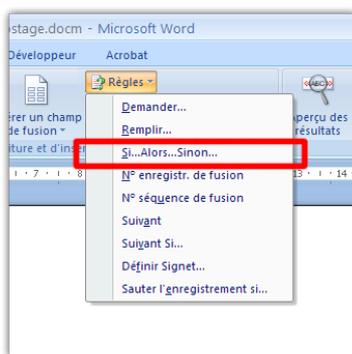
Le champ contenant votre invite et ses commutateurs



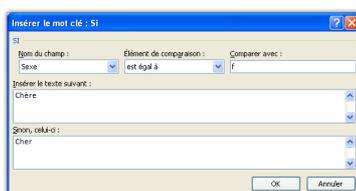
Le champ Remplir

6.1.3. Si Alors Sinon

Dans les règles, la règle la plus intéressante est **Si... Alors... Sinon**. Si vous souhaitez utiliser des articles ou qualificatifs pour différencier les hommes des femmes, par exemple Cher Marcel ou Chère Marie, vous devriez ajouter autant de champs dans votre base de données que de qualificatifs ou articles que vous pensez devoir utiliser. Cette règle assez simple permet de faire un choix en fonction d'un champ de votre base de données. Imaginez que vous ajoutiez le champ sexe dans vos données, vous pouvez faire ce choix à n'importe quel endroit de votre document.



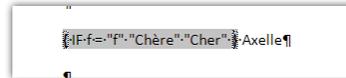
Vous obtiendrez une boîte de dialogue :



Et comme résultat :



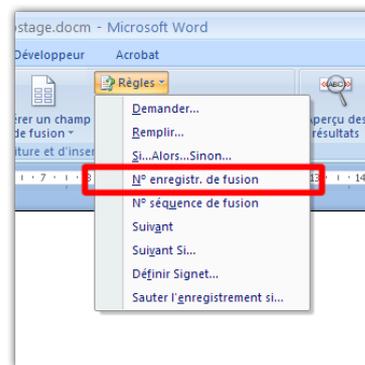
Le contenu du champ si vous l'affichez sera celui-ci



On se trouve avec un champ Si : [Lien 101](#)

6.1.4. Numéro d'enregistrement

Ce champ ajoute le numéro de l'enregistrement au document. Si vous désirez numéroter vos lettres, ce champ est tout indiqué.



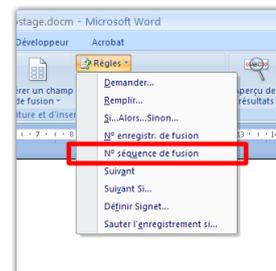
Numéro d'enregistrement



Le champ

6.1.5. Numéro de séquence

La valeur obtenue sera identique au champ Numéro d'enregistrement si vous utilisez tous les enregistrements. Si vous filtrez les enregistrements, ce numéro sera différent.



Numéro de séquence



Le champ

6.1.6. Suivant

Ce champ va passer à l'enregistrement suivant. C'est ce champ qui est utilisé par Word si vous utilisez le

publipostage pour la génération d'un répertoire.

Pour que cette commande fonctionne, elle ne peut être placée devant le premier champ de votre publipostage et elle n'agit que pour le champ qui la suit directement.

6.1.7. Suivant Si

Ce champ va fusionner dans le même document l'enregistrement suivant si la condition n'est pas remplie, si la condition est remplie, on passe alors au document suivant. On utilise ce champ lorsque plusieurs enregistrements composent les données à utiliser pour un même document de publipostage. C'est le cas pour des factures.

L'inconvénient, c'est qu'il est impératif de connaître le nombre d'occurrences que l'on va rencontrer. Nous verrons plus loin dans ce tutoriel comment utiliser une méthode bien plus souple en VBA.

6.1.8. Définir Signet

Ce champ permet d'affecter une valeur à un signet. Cette valeur peut être directement intégrée au champ.

```
{ SET Monsignet "MaValeur" }
```

Dans ce cas, le signet prendra la valeur "MaValeur". Pour afficher le contenu du signet, vous devrez utiliser un champ REF.

```
{ REF Monsignet }
```

On peut aussi lui affecter la valeur d'un champ de publipostage. Cette utilisation peut être très utile si vous avez un document de plusieurs pages et que vous souhaitez afficher au début de chaque page le nom du destinataire.

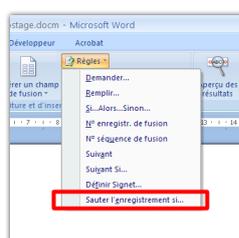
```
{ SET Monsignet { MERGEFIELD Nom } }  
{ REF Monsignet }
```

6.1.9. Sauter l'enregistrement

Ce champ permet d'éviter l'impression de certains enregistrements s'ils ne répondent pas à certaines conditions. L'exemple le plus simple est l'envoi de lettre uniquement aux femmes. Si c'est un homme, l'enregistrement doit être ignoré. La fonction de ce champ est identique au filtre.

Ne soyez pas surpris, cette clause de rejet n'affecte que le publipostage et pas son aperçu.

On obtient ce critère de sélection avec la méthode suivante :



Le résultat obtenu :



Si votre affichage ne ressemble pas à l'illustration, utilisez la combinaison de touche Alt + F9 pour basculer l'affichage des champs et n'oubliez pas de refaire cette opération pour annuler l'effet.

Le code :

```
{ SKIPIF { MERGEFIELD Sexe } = "H" }
```

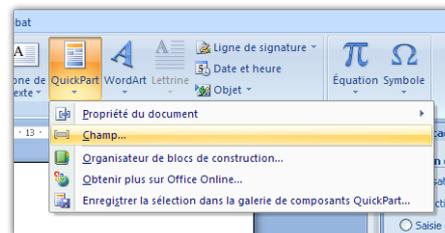
6.2. Les images

Dans l'utilisation du publipostage, on ne trouve pas souvent de renseignement sur l'affichage des images.

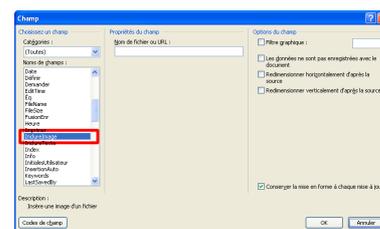
Il est possible d'obtenir pour chaque enregistrement une image différente, imaginez un répertoire où vous pourriez associer une photo à son propriétaire. Si vous décortiquez un champ d'insertion d'image dans Word, vous obtenez ceci :

```
{ INCLUDEPICTURE "c:\\temp\\a.jpg" \*  
MERGEFORMAT }
```

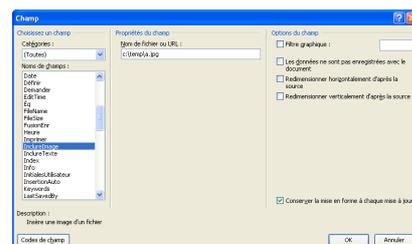
La méthode pour y parvenir est la suivante : dans l'onglet **Insérer**, le bouton **QuickPart** et la commande **Champ**.



Choisissez champ **Inclureimage**.



Et remplissez avec le chemin de votre image.



Il ne nous reste plus que l'utilisation d'un champ de publipostage pour remplir le champ **INCLUDEPICTURE**.

```
{ INCLUDEPICTURE { MERGEDFIELD Path } \*
MERGEFORMAT }
```

Les commutateurs utilisables sont identiques aux commutateurs des champs.

Le champ InclureImage : [Lien 102](#)

Lors de l'édition du champ, on se rend compte que le chemin du fichier doit comporter des doubles antislashes au lieu des simples antislashes. Pour obtenir une insertion d'image lors d'un publipostage, nous devons donc ajouter un champ dans l'enregistrement qui contiendra le chemin de l'image mais avec des doubles antislashes.

Le chemin normal

```
C:\Temp\a.jpg
```

Le chemin à utiliser

```
C:\\Temp\\a.jpg
```

L'insertion des images ne se produit que lors de la fusion, le mode prévisualisation ne permet pas l'affichage des images. Pour que toutes les images soient mises à jour, utilisez la combinaison **Ctrl + A** pour sélectionner tout le document et ensuite la touche **F9** pour mettre les champs à jour.

7. Le VBA

7.1. Introduction

La commande pour la fusion et le publipostage en Word est **MailMerge**, et l'exécution de la fusion avec **Execute**.

```
ActiveDocument.MailMerge.Execute
```

Pour que cette commande ne lève pas d'erreur, certaines conditions doivent être remplies. La principale : le document doit être un document principal de fusion et publipostage.

Un document principal de fusion est un document qui possède une source de données associée.

Pour savoir si votre document est un document principal de fusion, vous pouvez faire le test suivant :

```
Sub TestDocPrincipal()
If ActiveDocument.MailMerge.State = 2 Then MsgBox
"Le document est un document principal de
fusion !"
End Sub
```

Valeur	Nom	Description
0	wdNormalDocument	Le document n'est pas inclus dans un processus de fusion et publipostage.
1	wdMainDocumentOnly	Document de base sans données jointes.

2	wdMainAndDataSource	Document de base avec une source de données jointe.
3	wdMainAndHeader	Document de base avec un fichier d'en-tête.
4	wdMainAndSourceAndHeader	Document de base avec une source de données et un fichier d'en-tête joints.
5	wdDataSource	Source de données sans document de base.

Nous pourrions même envisager de faire une fonction pour tester l'état du document.

Routine d'appel

```
Sub TestEtatDoc()
MsgBox EtatDocument
(ActiveDocument.MailMerge.State)
End Sub
```

Fonction

```
Function EtatDocument( btEtat As Byte ) As String
Select Case btEtat
Case 0
EtatDocument = "Le document n'est pas
inclus dans un processus de fusion et
publipostage."
Case 1
EtatDocument = "Document de base sans
données jointes."
Case 2
EtatDocument = "Document de base avec une
source de données jointe."
Case 3
EtatDocument = "Document de base avec un
fichier d'en-tête."
Case 4
EtatDocument = "Document de base avec une
source de données et un fichier d'en-tête
joints."
Case 5
EtatDocument = "Source de données sans
document de base."
End Select
End Function
```

7.2. Ajout d'une source de données

Si votre document n'est pas un document principal de fusion, vous devez si vous désirez faire du publipostage lui adjoindre une source de données. La méthode pour l'ajout de données à votre document est **OpenDataSource**.

Ajout de données

```
Sub DocAjoutSource()
ActiveDocument.MailMerge.OpenDataSource .....
End Sub
```

7.2.1. Données SQL Server

Pour ajouter des données d'une base SQL Server, vous devez utiliser le chemin du fichier se trouvant dans le répertoire "Mes sources de données".

Si par curiosité vous ouvrez ce fichier "*.ODC" à l'aide du **Bloc-note**, vous verrez que vous êtes en présence d'un fichier XML contenant des données et des scripts.

Données SQL Server

```
Sub DocAjoutSource ()
ActiveDocument.MailMerge.OpenDataSource .....
End Sub
```

7.2.1. Données SQL Server

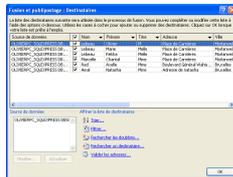
Pour ajouter des données d'une base SQL Server, vous devez utiliser le chemin du fichier se trouvant dans le répertoire "Mes source de données".

Si par curiosité vous ouvrez ce fichier "*.ODC" à l'aide du **Bloc-note**, vous verrez que vous êtes en présence d'un fichier XML contenant des données et des scripts.

Données SQL Server

```
Sub DocAjoutSource ()
ActiveDocument.MailMerge.OpenDataSource _
Name:="C:\Documents and
Settings\UserName\Mes documents\Mes sources de
données\OLIVIERPC_SQLEXPRESS DBSQL Table1.odc"
End Sub
```

Après exécution, vous pouvez vérifier que les données sont "liées" au document dans l'onglet publipostage et le bouton Modifier la liste des destinataires.



7.2.2. Données Access

La méthode pour lier des données Access ne diffère pas de la méthode utilisée pour des données SQL Server. Mais au lieu d'utiliser le fichier de connexion à SQL Server, nous allons utiliser le nom du fichier Access.

Données Access

```
Sub DocAjoutSource ()
ActiveDocument.MailMerge.OpenDataSource _
Name:="C:\Documents and
Settings\Olivier\Mes documents\Mes sources de
données\adresses.mdb"
End Sub
```

La même manipulation qu'au paragraphe 7-B-1, vous permettra de vérifier que les données sont liées au document.

7.2.3. Données Excel

Une fois de plus, la méthode pour utiliser une source de données basée sur un fichier Excel est la même que pour les autres fichiers de données.

Données Excel

```
Sub DocAjoutSource ()
ActiveDocument.MailMerge.OpenDataSource
Name:="C:\Documents and Settings\Olivier\Mes
documents\Mes sources de données\adresses.xlsx"
End Sub
```

Avec les mêmes manipulations, on peut vérifier que les données sont correctement liées au document.

7.3. Ajout d'un champ de données

Si vous n'avez pas défini de champ de sortie dans votre document, vous pouvez les insérer par le code. Quelle que soit la source choisie, la méthode pour insérer des champs reste la même.

Insertion d'un champ

```
ActiveDocument.Fields.Add Range:=Selection.Range,
Type:=wdFieldMergeField, Text:=""Nom""
```

La méthode pour insérer les champs et le texte est assez simple, il suffit d'écrire les lignes de code les unes sous les autres.

Insertion de champs et de texte

```
ActiveDocument.Fields.Add Range:=Selection.Range,
Type:=wdFieldMergeField, Text:=""Nom""
Selection.TypeText vbCrLf & " Je viens de mettre
le nom " & vbCrLf
ActiveDocument.Fields.Add Range:=Selection.Range,
Type:=wdFieldMergeField, Text:=""Prénom""
```

Vous avez aussi la possibilité d'insérer votre champ sur un signet.

Insérer sur signet

```
ActiveDocument.Fields.Add
Range:=Bookmarks(1).Range,
Type:=wdFieldMergeField, Text:=""Nom""
```

7.4. L'exécution

Il ne reste qu'à exécuter le publipostage. La méthode **Execute** va le faire et créer un nouveau document.

Exécution du publipostage

```
ActiveDocument.MailMerge.Execute
```

7.5. Destination du publipostage

Le publipostage permet d'envoyer le ou les documents vers une destination. Cette destination peut être un nouveau document, une imprimante, un Email ou un FAX.

```
ActiveDocument.MailMerge.Destination =
wdSendToNewDocument
```

Paramètre	Destination
wdSendToNewDocument	Le publipostage va se faire vers un nouveau document
wdSendToPrinter	Le résultat du publipostage sera directement envoyé sur l'imprimante

wdSendToEmail	Le résultat sera un envoi par mail du document
wdSendToFax	Le résultat du publipostage sera envoyé par FAX si un FAX est installé sur le PC.

Si vous utilisez wdSendToEmail ou wdSendToFax, vous devez vous assurer que le champ contenant l'adresse Email ou le numéro de fax est correctement renseigné.

7.6. Code Complet

Voilà le code complet des paragraphes précédents avec l'utilisation des signets.

Code Complet avec signets

```
Sub DocPubliposte()
ActiveDocument.MailMerge.OpenDataSource _
    Name:="C:\Documents and
Settings\Olivier\Mes documents\Mes sources de
données\OLIVIERPC_SQLEXPRESS DBSQL Table1.odc"

ActiveDocument.Fields.Add
Range:=Bookmarks(1).Range,
Type:=wdFieldMergeField, Text:=""Nom""
ActiveDocument.Fields.Add
Range:=Bookmarks(2).Range,
Type:=wdFieldMergeField, Text:=""Prénom""
ActiveDocument.MailMerge.Execute

End Sub
```

Ce code s'il est utilisé plusieurs fois à la suite va insérer plusieurs fois les champs de publipostage. Il est donc préférable de l'utiliser lors de la création d'un document. Pour les utilisations ultérieures du document, il est préférable de se contenter de l'Exécution.

7.7. Enregistrement de chaque document

Il arrive fréquemment que l'on cherche à enregistrer les documents issus du publipostage dans des fichiers séparés. Les méthodes offertes par Word au travers de l'interface graphique ne permettent pas d'avoir un document par enregistrement ou ligne de données.

En exécutant le publipostage par le code, vous pouvez enregistrer chaque document avec comme nom les données contenues dans votre base de données.

Le principe est assez simple, on va récupérer le nombre d'enregistrements contenus dans la source de données (iR) et dans une boucle, nous allons parcourir ces enregistrements les uns après les autres.

For i = 1 To iR

Pour utiliser un enregistrement précis, nous allons utiliser i pour définir le premier et le dernier enregistrement de la série, ce qui ne nous donnera qu'un seul enregistrement.

```
Sub TestPublipost()
' Déclaration des variables
Dim iR As Integer
Dim i As Integer
Dim oDoc As Document
```

```
Dim DocName As String
Dim oDS As MailMergeDataSource

' Affectation des objets
Set oDoc = ActiveDocument
Set oDS = oDoc.MailMerge.DataSource

iR = MailMerge.DataSource.RecordCount
Debug.Print iR
For i = 1 To iR
    With oDoc.MailMerge
        'Définition du premier et dernier
enregistrement
        .DataSource.FirstRecord = i

        .DataSource.LastRecord = i
        ' Envoi des données dans un nouveau
document
        .Destination = wdSendToNewDocument
        ' Exécution du publipostage
        .Execute
        ' Actualisation de l'enregistrement pour
la sauvegarde
        .DataSource.ActiveRecord = i
        'Utilisation de deux champs pour obtenir
le nom du document
        DocName = .DataSource.DataFields(2).Value
        DocName = DocName & "-" &
.DataSource.DataFields(3).Value
        Debug.Print DocName; i
    End With
    ' Sauvegarde du document publiposté
    With ActiveDocument
        .SaveAs "c:\temp\" & DocName & ".doc"
        .Close
    End With
Next i
End Sub
```

Les différents **Debug.Print** utilisés dans le code servent à "monitorer" le déroulement de la procédure.

La variable **DocName** va recevoir le nom de sauvegarde du document. Ce nom sera la concaténation de deux valeurs de champ, dans le code ci-dessus, nous utilisons la donnée du deuxième et du troisième champ.

DocName = .DataSource.DataFields(2).Value
DocName = DocName & "-" &
.DataSource.DataFields(3).Value

Finalement, c'est ce morceau de code qui va enregistrer le document dans le répertoire **C:\temp**

```
With ActiveDocument
    .SaveAs "c:\temp\" & DocName & ".doc"
    .Close
End With
```

8. Autre méthode VBA

La méthode VBA que nous venons de voir utilise les fonctionnalités du Publipostage de Word. Il est possible d'obtenir un résultat similaire sans utiliser la fonction Publipostage de Word.

Nous allons aborder ce sujet dans ce chapitre.

Nous allons aborder la connexion avec des données à un SQL server, la connexion avec des données Access et

terminer par la récupération de données d'un classeur Excel

8.1. Connexion à un SQL Server

Vous devez ajouter la référence **Microsoft ActiveX Data Object** à votre projet.

Pour les raisons énoncées un peu plus haut, j'ai utilisé SQL Server 2005 version Express.

La première chose à connaître est le nom de votre serveur SQL. Dans l'exemple qui suit, c'est le nom par défaut donné au serveur lors de son installation.

Le lien ci-dessous vous donne une vue d'ensemble sur le SQL Server 2005.

Présentation de SQL Server 2005 : [Lien 103](#)

Si vous suivez le lien ci-dessous, vous trouverez un article sur l'utilisation de Recordset ADO.

Comprendre les Recordset ADO : [Lien 104](#)

Le principe reste assez simple, il faut en premier lieu ouvrir une connexion vers le Serveur pour ensuite en extraire les données avec une commande SQL.

Connexion à un serveur SQL

```
Sub connexionSQL()  
Dim rs As ADO.DB.Recordset  
Dim stCon As ADO.DB.Connection  
Dim Utili As String  
Dim mdp As String  
Utili = ""  
mdp = ""  
  
'Création d'un objet ADO  
Set stCon = New ADO.DB.Connection  
'Accès aux données  
With stCon  
    .ConnectionString = "UID=" & Utili & ";PWD=" &  
    & mdp & ";" & "DRIVER={SQL Server};Server=" _  
    & "OLIVIERPC\SQLEXPRESS"  
    & ";Database=" & "DBSQL" & ";"  
    .Open  
End With  
  
'Création d'un objet Recordset (jeu  
d'enregistrements)  
Set rs = New ADO.DB.Recordset  
  
rs.Open "SELECT * FROM Table1", stCon  
'Boucle pour parcourir les enregistrements et  
afficher leur contenu  
'dans la fenêtre d'exécution.  
While Not rs.EOF  
Debug.Print rs.Fields(2).Value  
rs.MoveNext  
Wend  
  
'libération des objets  
stCon.Close  
set stCon = Nothing  
Set rs = Nothing  
  
End Sub
```

Si vous avez correctement utilisé le code, vous devriez avoir une liste avec le contenu du troisième champ de votre table.

8.2. Connexion à une base de données Access

Vous devez ajouter la référence **Microsoft DAO Object Library** à votre projet

Le principe pour accéder aux données contenues dans une base de données Access est sensiblement le même que pour un Serveur SQL avec du ADO.

Comme nous allons utiliser du DAO, je vais vous donner un lien sur l'utilisation du DAO.

Définition et manipulation de données avec DAO : [Lien 105](#)

Récupération des données Access

```
Sub ConnexionDBAccess()  
'Déclaration des variables  
Dim db As DAO.Database  
Dim rs As DAO.Recordset  
Dim sql As String  
'Affectation de la sentence SQL  
sql = "SELECT * From T_Adresses"  
'Affectation des Objets d'accès aux données  
Set db = DBEngine.OpenDatabase("C:\Documents and  
Settings\Olivier\Mes documents\Mes sources de  
données\DB_Publipostage.mdb")  
Set rs = db.OpenRecordset(sql)  
'Boucle pour la récupération des données  
While Not rs.EOF  
Debug.Print rs.Fields(2)  
rs.MoveNext  
Wend  
  
'Libération des objets  
rs.Close  
db.Close  
  
Set rs = Nothing  
Set db = Nothing  
  
End Sub
```

Comme pour le paragraphe précédent, vous devriez avoir une liste avec le contenu du troisième champ de votre table.

8.3. Récupération de données d'une feuille Excel

Vous devez ajouter la référence **Microsoft Excel Object Library** à votre projet

Comparée aux deux méthodes précédentes, la récupération de données contenues dans un classeur Excel est un peu différente.

Nous allons ouvrir le classeur, obtenir le nombre de lignes (*iR*) et le nombre de colonnes (*iC*) pour ensuite dans une double boucle récupérer les données des cellules en fonction de leur adresse.

Extraction des données d'une feuille Excel

```
Sub donneeAvecExcel()  
'Déclaration des variables  
Dim xlApp As Excel.Application  
Dim xlWb As Excel.Workbook  
Dim xlSh As Excel.Worksheet  
Dim iR As Integer  
Dim iC As Integer  
Dim i As Integer, j As Integer
```

```
'Affectation des données aux variables
Set xlApp = New Excel.Application
Set xlWb = xlApp.Workbooks.Open("C:\Documents and
Settings\Olivier\Mes documents\Mes sources de
données\adresses.xlsx")
Set xlSh = xlWb.Worksheets(1)
'Récupération du nombre de lignes et de colonnes
iR = xlSh.UsedRange.Rows.Count
iC = xlSh.UsedRange.Columns.Count
'Boucle pour adresser les cellules contenant des
données
'en excluant la première ligne qui contient les
titres
For i = 2 To iR
  For j = 1 To iC
    Debug.Print xlSh.Cells(i, j)
  Next j
Next i

xlWb.Close
xlApp.Quit
Set xlSh = Nothing
Set xlWb = Nothing
Set xlApp = Nothing

End Sub
```

8.4. Utilisation des données dans un document

Maintenant que nous avons vu comment extraire des données, nous allons les utiliser. La manipulation reste la même, que les données soient issues d'une base de données SQL Server, d'une base de données Access ou d'une feuille Excel.

Le principe que nous allons mettre en œuvre est relativement simple, les données récupérées seront envoyées vers des signets que nous aurons mis dans notre document. L'utilisation de signets est la plus souple, elle permet d'insérer les données à des endroits précis de notre document.

Pour insérer des données sur un signet, c'est assez simple :

Insertion de données sur un signet

```
ActiveDocument.Bookmarks(1).Range.Text = "La
Donnée"
```

Le principe est relativement simple, dans l'exemple que je viens de donner, la donnée est fixe. Mais on peut utiliser une donnée extraite :

```
ActiveDocument.Bookmarks(1).Range.Text =
xlSh.Cells(i, j)
```

Voilà pour l'insertion de données.

8.4.1. Cas simple

Nous allons distinguer le cas simple (un enregistrement par document : une lettre) du cas multiple (plusieurs enregistrements par document : une facture) . Cet enregistrement peut contenir le nom du fichier, son répertoire de sauvegarde, etc.

Pour les données, il suffit de les transférer vers le document dans des signets.

Dans les paragraphes précédents, nous avons vu comment extraire les données. Nous allons aller un peu plus loin et traiter les données.

Traitement des données

```
While Not rs.EOF
'Traitement des données
rs.MoveNext
Wend
```

Nous allons remplir cette boucle avec le traitement des données. Nous allons utiliser les champs pour les envoyer dans le document.

Pour le code, je vais utiliser la connexion à une base de données Access. Je suppose que pour chaque document, il faut utiliser le nom issu des données et la date au format "aa-mm-jj" comme nom de document

Le code avec le transfert de données

```
Sub ConnexionDBAccess()
'Déclaration des variables
Dim db As DAO.Database
Dim rs As DAO.Recordset
Dim oDoc As Document
Dim sql As String
'Affectation de la sentence SQL
sql = "SELECT * From T_Adresses"
'Affectation des Objets d'access aux données
Set db = DBEEngine.OpenDatabase("C:\Documents and
Settings\Olivier\Mes documents\Mes sources de
données\DB_Publipostage.mdb")
Set rs = db.OpenRecordset(sql)
'Boucle pour la récupération des données
While Not rs.EOF
'Affectation d'un objet document à la variable
document
Set oDoc = Documents.Add("C:\Documents and
Settings\Olivier\Application
Data\Microsoft\Templates\pub.dotm")
'Ecriture des données sur les signets
oDoc.Bookmarks(1).Range.Text = rs.Fields(1)
oDoc.Bookmarks(2).Range.Text = rs.Fields(2)
oDoc.Bookmarks(3).Range.Text = rs.Fields(3)
'Sauvegarde du document
oDoc.SaveAs "c:\temp\" & rs.Fields(1) & "-" &
Format(Date, "yy-mm-dd") & ".docx"
'Libération de l'objet document
oDoc.Close
Set oDoc = Nothing

rs.MoveNext
Wend
'Libération des objets
rs.Close
db.Close

Set rs = Nothing
Set db = Nothing

End Sub
```

Dans l'exemple ci-dessus, j'ai utilisé les numéros des champs et des signets par facilité, mais il est possible d'utiliser leur nom.

Le traitement avec une feuille Excel

```
Sub donneeAvecExcel()  
'Déclaration des variables  
Dim xlApp As Excel.Application  
Dim xlWb As Excel.Workbook  
Dim xlSh As Excel.Worksheet  
Dim iR As Integer  
Dim i As Integer, j As Integer  
Dim oDoc As Document  
  
'Affectation des données aux variables  
Set xlApp = New Excel.Application  
Set xlWb = xlApp.Workbooks.Open("C:\Documents and  
Settings\Olivier\Mes documents\Mes sources de  
données\adresses.xlsx")  
Set xlSh = xlWb.Worksheets(1)  
'Récupération du nombre de lignes et de colonnes  
iR = xlSh.UsedRange.Rows.Count  
  
' Récupération des données de la feuille pour les  
injecter dans le document.  
For i = 2 To iR  
    Debug.Print xlSh.Cells(iR, 2); iR  
    Set oDoc = Documents.Add("C:\Documents  
and Settings\Olivier\Application  
Data\Microsoft\Templates\pub.dotm")  
    oDoc.Bookmarks(1).Range.Text =  
xlSh.Cells(i, 1)  
    oDoc.Bookmarks(2).Range.Text =  
xlSh.Cells(i, 2)  
    oDoc.Bookmarks(3).Range.Text =  
xlSh.Cells(i, 3)  
    oDoc.SaveAs "c:\temp\" & xlSh.Cells(i, 2)  
& "-" & Format(Date, "yy-mm-dd") & ".docx"  
    oDoc.Close  
    Set oDoc = Nothing  
Next i  
  
xlWb.Close  
xlApp.Quit  
Set xlSh = Nothing  
Set xlWb = Nothing  
Set xlApp = Nothing  
  
End Sub
```

Dans le premier code utilisé avec une feuille Excel, nous avons tenu compte du nombre de colonnes. Lors du publipostage, ce nombre de colonnes n'est plus utilisé.

Il est bien évidemment possible d'imprimer le document avant la sauvegarde.

Pour l'impression

```
oDoc.SaveAs "c:\temp\" & xlSh.Cells(i, 2) & "-" &  
Format(Date, "yy-mm-dd") & ".docx"  
oDoc.PrintOut  
oDoc.Close
```

8.4.2. Cas Multiple

Par cas multiple, je suppose plusieurs enregistrements ou lignes pour un même document, comme pour une facture. On peut y arriver par les champs de fusion et de publipostage, mais avec certaines restrictions, on doit connaître le nombre maximum d'occurrences avant fusion. Il n'est pas possible d'insérer les données dans une table.

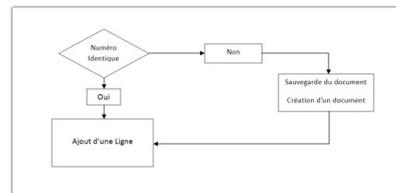
Pour la récupération des données, le principe reste identique, ce qui change, c'est le traitement des données. Dans cette partie nous allons simplement nous attarder sur le traitement des données et non plus sur leur récupération. Pour chaque enregistrement ou ligne, nous allons vérifier si une des valeurs est identique à la ligne précédente on ajoute une ligne au document, si la valeur est différente, on crée un nouveau document et on ferme le précédent. Vous l'aurez compris, nous allons utiliser des structures conditionnelles dans notre traitement.

Voilà un exemple de données

IdFact	NoFacture	IdClient	Produit	Quantité
1	1	1	Ballon	10
2	1	1	Filet	1
3	2	1	Vélo	1
4	2	1	Casque	1
5	3	2	Ballon	5

Créez un modèle contenant des signets (au moins 3) et un tableau avec une ligne et deux colonnes.

Dans notre traitement, nous allons comparer les numéros de facture qui devraient être le critère de regroupement.



Publipostage Multiple

```
Sub donneeAvecExcel()  
On Error GoTo GestErr  
'Déclaration des variables  
Dim xlApp As Excel.Application  
Dim xlWb As Excel.Workbook  
Dim xlSh As Excel.Worksheet  
Dim iR As Integer  
Dim i As Integer, j As Integer  
Dim oDoc As Document  
Dim NoFact As Integer  
Dim oTbl As Table  
Dim stDocName As String  
  
'Affectation des données aux variables  
Set xlApp = New Excel.Application  
Set xlWb = xlApp.Workbooks.Open("C:\Documents and  
Settings\Olivier\Mes documents\Mes sources de  
données\adresses.xlsx")  
Set xlSh = xlWb.Worksheets(2)  
'Récupération du nombre de lignes et de colonnes  
iR = xlSh.UsedRange.Rows.Count  
NoFact = 0  
  
' Récupération des données de la feuille pour  
les injecter dans le document.
```

```

For i = 2 To iR
  If NoFact <> xlSh.Cells(i, 2) Then
    stDocName = "c:\temp\" & xlSh.Cells(i, 2)
    & "-" & Format(Date, "yy-mm-dd") & ".docm"
    oDoc.Close
    Set oDoc = Documents.Add("C:\Documents
and Settings\Olivier\Application
Data\Microsoft\Templates\publ.dotm")
    oDoc.Bookmarks(1).Range.Text =
xlSh.Cells(i, 1)
    oDoc.Bookmarks(2).Range.Text =
xlSh.Cells(i, 2)
    oDoc.Bookmarks(3).Range.Text =
xlSh.Cells(i, 3)
    Set oTbl = oDoc.Tables(1)
    oTbl.Rows.Add
    oTbl.Rows.Last.Cells(1).Range.Text =
xlSh.Cells(i, 4)
    oTbl.Rows.Last.Cells(2).Range.Text =
xlSh.Cells(i, 5)
    Set oTbl = Nothing
    oDoc.SaveAs stDocName
    'Affectation du nouveau numéro de facture
pour la comparaison
    NoFact = xlSh.Cells(i, 2)
  Else
    Set oTbl = oDoc.Tables(1)
    oTbl.Rows.Add
    oTbl.Rows.Last.Cells(1).Range.Text =
xlSh.Cells(i, 4)
    oTbl.Rows.Last.Cells(2).Range.Text =
xlSh.Cells(i, 5)
    Set oTbl = Nothing
    oDoc.Save
  End If
Next i

oDoc.Close

Set oDoc = Nothing
GestErr:
'Si pas de document ouvert on fait un resume next
If Err.Number = 91 Then Resume Next
Debug.Print "Erreur : " & Err.Number &
Err.Description
xlWb.Close
xlApp.Quit
Set xlSh = Nothing
Set xlWb = Nothing
Set xlApp = Nothing

End Sub

```

Si nous détaillons un peu le code.

```
oDoc.Close
```

On ferme le document, si aucun document n'est affecté à la variable oDoc, on déclenche une erreur. Cette erreur est gérée par :

```
On Error GoTo GestErr
```

et par :

```
GestErr:
'Si pas de document ouvert on fait un resume next
If Err.Number = 91 Then Resume Next
```

En effet, si aucun objet n'est affecté, alors une erreur 91 intervient. On en connaît la raison, on peut donc continuer la procédure avec **Resume Next**.

```
Set oDoc = Documents.Add("C:\Documents and
Settings\Olivier\Application
Data\Microsoft\Templates\publ.dotm")
oDoc.Bookmarks(1).Range.Text = xlSh.Cells(i, 1)
oDoc.Bookmarks(2).Range.Text = xlSh.Cells(i, 2)
oDoc.Bookmarks(3).Range.Text = xlSh.Cells(i, 3)
```

Cette partie a déjà été utilisée plus tôt !

```
Set oTbl = oDoc.Tables(1)
oTbl.Rows.Add
oTbl.Rows.Last.Cells(1).Range.Text =
xlSh.Cells(i, 4)
oTbl.Rows.Last.Cells(2).Range.Text =
xlSh.Cells(i, 5)
Set oTbl = Nothing
```

Dans ce petit morceau, on affecte à la variable Table le premier tableau du document. Nous allons ajouter une ligne à cette table et remplir cette ligne avec certaines données.

```
Else
  Set oTbl = oDoc.Tables(1)
  oTbl.Rows.Add
  oTbl.Rows.Last.Cells(1).Range.Text =
xlSh.Cells(i, 4)
  oTbl.Rows.Last.Cells(2).Range.Text =
xlSh.Cells(i, 5)
  Set oTbl = Nothing
  oDoc.Save
End If
```

Le code est utilisé deux fois, la seconde est utilisée pour n'ajouter des données qu'à la table et plus dans le corps du document.

Pour les fichiers ci-dessous, il suffit de mettre tous les fichiers dans le même répertoire. La macro est contenue dans le fichier "DocumentPrincipale". Téléchargez des fichiers exemples : [Lien 106](#)

9. Liens

Le lien suivant vous conduit vers un tutoriel sur le publipostage, mais au lieu d'utiliser Word, on utilise Access : [Lien 107](#)

Un tutoriel sur le publipostage avec Excel : [Lien 108](#)

L'utilisation du DAO : Définition et manipulation de données avec DAO ([Lien 109](#))

L'utilisation de l'ADO : [Lien 110](#)

10. Conclusion

Nous venons d'aborder quelques méthodes utilisées pour le publipostage sous Word. J'espère que vous pourrez y trouver une solution qui vous convient.

Retrouvez l'article d'Olivier Lebeau en ligne : [Lien 111](#)

Gérer la navigation dans un diaporama PowerPoint

Cet article a pour but de vous montrer comment gérer la navigation dans un diaporama PowerPoint.

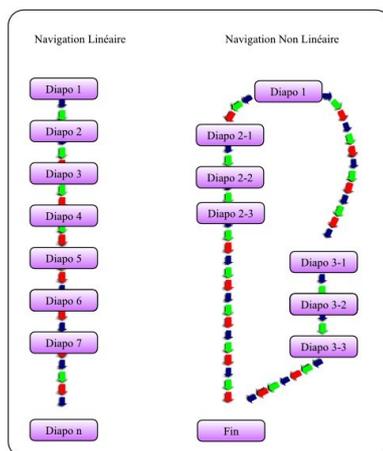
1. INTRODUCTION

Le but de cet article est de vous montrer comment on peut naviguer entre les diapositives dans PowerPoint.

Nous pouvons distinguer deux types de navigations :

- **la navigation linéaire** : on commence par la première diapositive, et on finit par la dernière ;
- **une navigation non linéaire** : l'ensemble des diapositives est composé en bloc et on passe d'un bloc à l'autre.

Le schéma ci-dessous illustre cela.



2. NAVIGATION AUTOMATIQUE

Par défaut, la transition d'une diapositive à l'autre se fait lors du clic sur le bouton de la souris.

Cependant, cette méthode a pour inconvénients :

- il faut que toutes les animations de la diapositive se soient déroulées pour le passage à la diapositive suivante ;
- votre diaporama est linéaire et vous ne pouvez imposer le passage vers une diapositive qui ne se trouve pas dans la suite logique ;
- l'inconvénient également d'une navigation automatique est que le timing entre les slides (diapositives) doit être impeccable, le lecteur doit avoir suffisamment de temps pour lire les infos mais le slide ne doit pas être trop long non plus.

2.1. Paramétrage manuel

Le paramétrage par défaut est heureusement modifiable.

- Affichez le volet Office (Menu Affichage/Volet Office).
 - Sélectionnez transition.
- Vous trouverez en bas de celui-ci l'image suivante :



- **Manuellement** : si cette case est cochée, cela indique que la transition entre les diapositives se fera manuellement, sinon, il faudra gérer le passage d'une diapositive à l'autre.
- **Automatiquement après** : permet de déterminer une temporisation entre chaque diapositive.
- **Appliquer à toutes les diapositives** : cela vous permet d'affecter le choix à l'ensemble de votre diaporama ou non.

Dans la suite de cet article nous allons décocher la case **Manuellement**, car nous voulons gérer nous même le passage d'une diapositive à l'autre.

PowerPoint 2007

Pour PowerPoint 2007, allez dans le menu animation pour voir apparaître le ruban animation, vous trouverez dans celui-ci les mêmes paramétrages.



2.2. Paramétrage par le code

Il est possible de programmer l'annulation de la transition manuelle entre les diapositives.

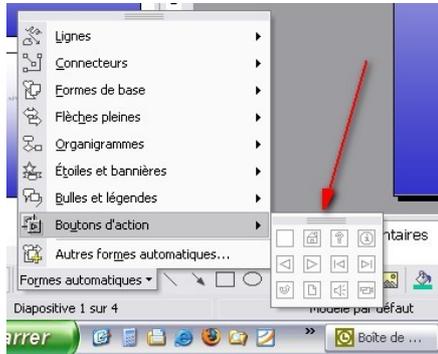
Le code suivant permet de désactiver la transition manuelle dans un diaporama.

```
VBA
Dim sld As Slide
For Each sld In ActivePresentation.Slides
    sld.SlideShowTransition.AdvanceOnClick = msoFalse
Next sld
```

3. LES BOUTONS D'ACTION

Les boutons d'action sont des boutons prédéfinis qui vont vous permettre de gérer la navigation entre vos diapositives.

Ceux-ci se trouvent dans la barre d'outils dessin sous PPT 2002 et 2003.



Vous pouvez également utiliser le menu Diaporama/Boutons d'action.

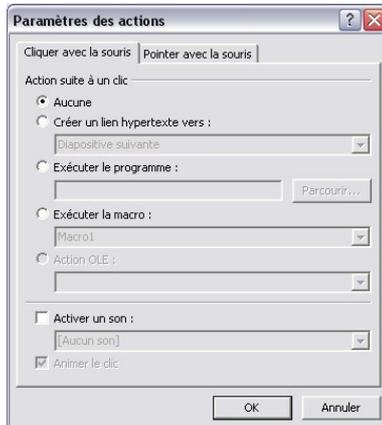
Sous PPT 2007 :

Activer le menu Insertion, dans le ruban sélectionner Formes, et ceux-ci se trouvent en bas de la liste déroulante.

3.1. Détail des boutons d'action

PowerPoint donne la possibilité d'utiliser 12 boutons d'actions.

Lorsque vous positionnez un bouton d'action sur une diapositive une fenêtre de paramétrage s'ouvre.



Cette fenêtre va vous permettre de paramétrer l'action du bouton, cependant, certains sont déjà préparamétrés.

Il peut y avoir deux types d'actions : sur le clic du bouton ou sur le survol par le curseur de la souris (vous trouverez le même type d'action pour ces deux événements).

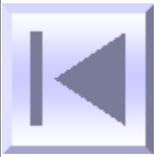
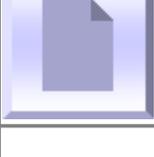
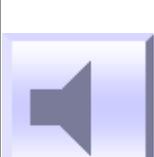
- **Aucune** : aucune action ne sera effectuée
- **Créer un lien hypertexte vers** : vous permettra de créer un lien hypertexte vers une autre diapositive ou une url. (Pour certains boutons d'action, cette propriété est définie par défaut.)
- **Exécuter le programme** : va vous permettre de lancer un programme (.exe).

- **Exécuter la macro** : va vous permettre d'exécuter une macro VBA.
- **Action Ole** : n'est pas utilisée pour les boutons d'action.
- **Activer un son** : va vous permettre de lancer un son lors d'une action sur le bouton (le son doit d'abord être inséré dans la diapositive pour apparaître dans la liste).

Vous pourrez ouvrir cette fenêtre à tout moment en effectuant un clic droit sur la souris (menu contextuel) et en sélectionnant Paramètres des Actions.

Le tableau suivant vous présentera les 12 boutons d'action.

Bouton d'Action	Image	Description
Bouton Personnalisé		Ce bouton va vous permettre de personnaliser une action grâce à la fenêtre de paramétrage des actions. Ce bouton ne comportant pas d'icône, il est possible d'écrire du texte à l'intérieur, pour cela afficher le menu contextuel (clic droit), et sélectionnez Modifier Texte. Pour que le texte s'adapte au bouton ou l'inverse, dans le menu contextuel sélectionnez Format de la Zone automatique, allez dans l'onglet Zone de texte, et sélectionnez ce que vous souhaitez (renvoie à la ligne, ajustement).
Bouton Accueil		Ce bouton préparamétré permet le retour à la première diapositive de votre diaporama
Bouton Aide		Ce bouton n'a aucune action prédéfinie, il peut vous permettre d'afficher une aide (lancement d'un UserForm par Macro par exemple).
Bouton Information		Ce bouton n'a aucune action prédéfinie, il peut vous permettre d'afficher une information (lancement d'un UserForm par Macro par exemple).
Bouton Précédent		Ce bouton préparamétré permet de revenir sur la diapositive précédente.

Bouton Suivant		Ce bouton préparamétré permet de passer à la diapositive suivante.
Bouton Début		Ce bouton préparamétré permet de revenir à la première diapositive. Il a le même effet que le bouton Accueil.
Bouton Fin		Ce bouton préparamétré permet d'aller à la dernière diapositive.
Bouton Retour		Ce bouton préparamétré permet de revenir à la dernière diapositive affichée. Ne pas confondre avec le bouton précédent, qui revient à la diapositive qui se trouve dans l'ordre du diaporama.
Bouton Document		Ce bouton n'est pas préparamétré, utilisez-le pour lancer un programme ou visualiser un autre document.
Bouton Son		Ce bouton est préparamétré, pour vous permettre de lancer un son. Une liste de sons prédéfinis apparaît, mais vous pouvez toujours en jouer d'autres (allez en bas de la liste déroulante).
Bouton Video		Ce bouton n'est pas préparamétré, utilisez-le pour lancer une vidéo.

3.2. Affecter les boutons d'action à toutes les diapositives

Si vous avez plusieurs diapositives et que vous voulez ajouter à celles-ci les boutons "Suivant", "Précédent" et "Accueil", cela devient fastidieux de le faire pour chaque diapositive.

Nous allons voir différentes manières de réaliser simplement cette opération.

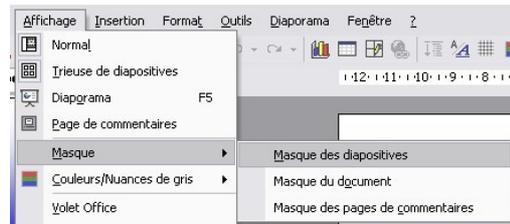
3.2.1. Manuellement

Il est donc possible d'affecter manuellement nos boutons d'action à chaque diapositive.

Pour effectuer cette opération nous allons utiliser un masque de diapositive.

Allez dans le menu **Affichage/Masque/Masque de**

diapositive.



Ensuite positionnez vos trois boutons d'action.

Renommez le masque PageNormal à l'aide du cinquième bouton de la barre d'outils.

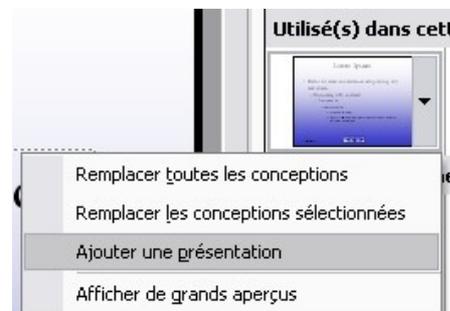
Puis cliquez sur Conserver le masque et ensuite cliquez sur Désactiver le mode Masque.



Toutes vos diapositives comporteront maintenant vos trois boutons d'action.

Cependant, l'inconvénient de cette méthode est que la première diapositive comporte les boutons Précédent et Accueil. Nous allons voir comment modifier cela.

- Repassez en mode diapositive.
- Sélectionnez la diapositive dans le volet Office.
- Sélectionnez **Ajouter une présentation**.



Vous verrez alors une deuxième présentation identique à la première s'afficher à la droite de celle-ci.

- Sélectionnez-la.
- Passez en mode masque de diapositive.
- Enlevez les boutons Précédent et Accueil.

Puis renommez-la en PremièrePage, et désactivez le mode masque.

Mais vous verrez que toutes les diapositives ont pris le nouveau masque.

Pour rétablir l'ensemble :

- Sélectionnez le premier masque et cliquez sur Appliquer à toutes les diapositives.
- Sélectionnez la première diapositive, allez dans le dernier masque (PremièrePage) et sélectionnez

Appliquer aux diapositives sélectionnées.

Vous avez maintenant la première diapositive avec le bouton Suivant, et les autres avec tous les boutons.

De la même manière vous pouvez faire le masque de la dernière diapositive.

3.2.2. Par programmation

L'ajout de boutons d'action peut également être fait grâce à du code VBA, nous allons donc créer nos boutons par programmation et les positionner.

Dans un module, on positionne en entête la déclaration des variables :

```
VBA
' ===== variables du module =====
Dim sld As Slide
Dim i As Integer
Dim shp As Shape
Dim intTopBtn As Integer ' détermine la position
des boutons par rapport à la hauteur
Dim intHeightBtn As Integer ' détermine la
hauteur de nos boutons
Dim intWidthBtn As Integer ' détermine la largeur
de nos boutons
```

Puis toujours dans ce module on positionne les routines suivantes :

```
VBA
Public Sub BtnPrecedent()
' =====
' Procédure : Module :mod_BoutonAction //
Procédure : BtnPrecedent
' DateTime : 07/03/2008 05:06
' Auteur : Starec - Philippe JOCHMANS -
http://starec.developpez.com
' Description : Cette procédure crée le
bouton précédent
' =====

' ===== variables =====
Dim intLeft As Integer ' position du bouton
par rapport au bord gauche
intLeft =
(ActivePresentation.PageSetup.SlideWidth / 2) -
(intWidthBtn * 1.5)

' ajout du bouton et positionnement

Set shp =
sld.Shapes.AddShape(msoShapeActionButtonBackorPre
vious, intLeft, intTopBtn, intWidthBtn,
intHeightBtn)
' action du bouton, et mise en place de la
couleur
With shp
.ActionSettings(ppMouseClick).Action =
ppActionPreviousSlide
.Fill.ForeColor.RGB = RGB(200, 180, 250)
.Name = "Precedent"
End With
End Sub

Public Sub BtnAccueil()
' =====
' Procédure : Module :mod_BoutonAction //
Procédure : BtnAccueil
' DateTime : 07/03/2008 05:06
```

```
' Auteur : Starec - Philippe JOCHMANS -
http://starec.developpez.com
' Description : Cette procédure crée le
bouton précédent
' =====

' ===== variables =====
Dim intLeft As Integer ' position du bouton
par rapport au bord gauche
intLeft =
(ActivePresentation.PageSetup.SlideWidth / 2) -
(intWidthBtn / 2)

' ajout du bouton et positionnement

Set shp =
sld.Shapes.AddShape(msoShapeActionButtonHome,
intLeft, intTopBtn, intWidthBtn, intHeightBtn)
' action du bouton, et mise en place de la
couleur
With shp
.ActionSettings(ppMouseClick).Action =
ppActionFirstSlide
.Fill.ForeColor.RGB = RGB(200, 180, 250)
.Name = "Accueil"
End With
End Sub

Public Sub BtnSuivant()
' =====
' Procédure : Module :mod_BoutonAction //
Procédure : BtnSuivant
' DateTime : 07/03/2008 05:06
' Auteur : Starec - Philippe JOCHMANS -
http://starec.developpez.com
' Description : Cette procédure crée le
bouton précédent
' =====

' ===== variables =====
Dim intLeft As Integer ' position du bouton
par rapport au bord gauche
intLeft =
(ActivePresentation.PageSetup.SlideWidth / 2) +
(intWidthBtn / 2)

' ajout du bouton et positionnement

Set shp =
sld.Shapes.AddShape(msoShapeActionButtonForwardor
Next, intLeft, intTopBtn, intWidthBtn,
intHeightBtn)
' action du bouton, et mise en place de la
couleur
With shp
.ActionSettings(ppMouseClick).Action =
ppActionNextSlide
.Fill.ForeColor.RGB = RGB(200, 180, 250)
.Name = "Suivant"
End With
End Sub
```

Ces trois procédures ont la même structure :

- on crée un bouton avec la méthode **AddShapes**, on détermine le type de bouton d'action, on le positionne ;
- ensuite on détermine l'action du bouton, sa couleur, et son nom.

Pour lancer la création de ceux-ci dans l'ensemble du diaporama on ajoute cette procédure :

VBA

```
Public Sub AjoutBoutonAction()  
' =====  
' Procedure : Module :mod_BoutonAction //  
Procédure : AjoutBoutonAction  
' DateTime : 05/03/2008 05:45  
' Auteur : Starec - Philippe JOCHMANS -  
http://starec.developpez.com  
' Description : Cette procédure va permettre  
d'ajouter des boutons d'action à notre  
' diaporama  
' =====  
  
' ===== affectation =====  
' on récupère le nombre de slides  
(diapositives)  
i = ActivePresentation.Slides.Count  
' pour positionner le haut de nos boutons  
intTopBtn =  
ActivePresentation.PageSetup.SlideHeight - 100  
' dimensions des boutons  
intHeightBtn = 50  
intWidthBtn = 50  
  
For Each sld In ActivePresentation.Slides  
Select Case sld.SlideIndex  
Case 1  
' on est sur la première diapositive  
Call BtnSuivant  
Case i  
' on est sur la dernière diapositive  
Call BtnPrecedent  
Call BtnAccueil  
Case Else  
' on est sur les autres diapositives  
Call BtnPrecedent  
Call BtnAccueil  
Call BtnSuivant  
End Select  
Next sld  
End Sub
```

Le principe est simple, on parcourt la liste des **slides**, et selon le cas on ajoute les boutons adéquats.

Positionnez-vous sur cette procédure et appuyez sur la touche F5, vos diapositives possèdent dorénavant des boutons de navigation.

Si vous devez ajouter une diapositive à votre diaporama, il va donc falloir supprimer tous les boutons pour les recréer par la suite.

Pour cela, positionnez cette routine dans votre module.

VBA

```
Public Sub SupBoutonAction()  
' =====  
' Procedure : Module :mod_BoutonAction //  
Procédure : SupBoutonAction  
' DateTime : 05/03/2008 06:16  
' Auteur : Starec - Philippe JOCHMANS -  
http://starec.developpez.com  
' Description : Cette procédure va permettre  
de supprimer les boutons d'action  
' =====  
  
' ===== déclaration =====  
Dim sld As Slide
```

```
Dim shp As Shape  
Dim i As Integer  
For i = 1 To 3  
For Each sld In ActivePresentation.Slides  
For Each shp In sld.Shapes  
If shp.Name = "Precedent" Or  
shp.Name = "Accueil" Or shp.Name = "Suivant" Then  
shp.Delete  
End If  
Next shp  
Next sld  
Next i  
End Sub
```

Exécutez-la, tous vos boutons seront supprimés, ensuite relancez la routine de création des boutons. Votre diaporama est maintenant à nouveau opérationnel.

N'oubliez pas de décocher la case de parcours manuel (voir plus haut), ou d'inclure dans la création de vos boutons le code qui permet d'effectuer cette opération.

Toutes ces procédures ne sont valables que lors d'un parcours linéaire de votre diaporama.

3.3. Formatage des boutons d'action

Il n'est pas possible de modifier l'icône des boutons d'action.

Vous pouvez cependant :

- modifier la couleur comme n'importe quelle forme grâce à la barre d'outils dessin ou par l'intermédiaire du menu contextuel en sélectionnant Format de la zone automatique ;
- ajouter du texte, mais l'icône se trouvera toujours centrée, à réserver au bouton personnalisé ;
- modifier la profondeur du bouton en agissant sur le losange jaune qui apparaît lorsque vous avez sélectionné le bouton (l'icône sera redimensionnée en conséquence).

4. LES IMAGES ET LES FORMES

À la place des boutons d'action vous pouvez créer vous-même vos propres boutons de navigation à l'aide d'images ou de formes.

Nous allons étudier le cas des formes, en sachant que pour les images le fonctionnement est similaire.

4.1. Manuellement

Le positionnement manuel est similaire au positionnement des boutons d'action, avec l'utilisation des masques de diapositives.

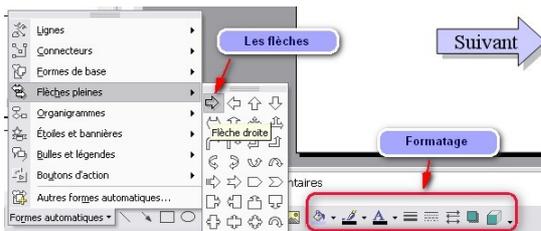
4.1.1. Création des formes

Nous allons juste voir comment créer nos formes par un exemple.

Création des flèches :

Vous pouvez utiliser n'importe quel type de forme, mais graphiquement le plus parlant ce sont des flèches.

Vous trouverez ci-dessous la procédure pour créer une flèche :



Vous pourrez bien sûr formater celle-ci à votre goût.

Pour ajouter du texte à celle-ci, faites un clic droit sur la flèche pour visualiser le menu contextuel et sélectionnez **Ajouter Texte**.

Effectuez la même opération pour la flèche de retour, et vous pouvez utiliser une autre forme pour l'accueil.

4.1.2. Les actions sur les formes

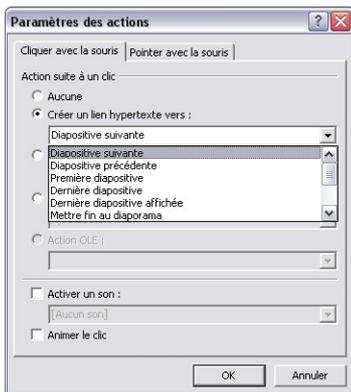
Contrairement aux boutons d'action, les formes ne sont pas préparamétrées, il vous appartient donc de réaliser cette opération.

Pour cela, faites un clic droit sur la forme pour faire apparaître le menu contextuel, et sélectionnez **Paramètres des actions**.

Vous arrivez sur l'onglet "Cliquer avec la souris", sélectionner **Créer un lien hypertexte vers**, et vous aurez une liste vous donnant plusieurs choix.

Il ne vous reste plus qu'à faire votre choix, et de cliquer sur OK en bas de la fenêtre.

Effectuez le paramétrage pour chacun de vos boutons.



En utilisant la méthode évoquée sur les masques de diapositives, vous pourrez créer un masque pour chaque type.

La manipulation pour affecter l'action doit se faire lorsque vous êtes en mode masque pour que celle-ci soit opérationnelle.

4.2. Par programmation

Le principe sera exactement le même que pour la création des boutons d'action. La seule différence concernera la

création de la forme par d'autres paramètres.

Déclaration des variables :

```
VBA
' ===== variables du module =====
Dim sld As Slide
Dim i As Integer
Dim shp As Shape
Dim intTopBtn As Integer ' détermine la position
des boutons par rapport à la hauteur
Dim intHeightBtn As Integer ' détermine la
hauteur de nos boutons
Dim intWidthBtn As Integer ' détermine la largeur
de nos boutons
```

Ajout des Formes :

```
Public Sub AjoutFormes ()
' =====
' Procédure : Module :mod_BoutonForm //
Procédure : AjoutFormes
' DateTime : 08/03/2008 07:44
' Auteur : Starec - Philippe JOCHMANS -
http://starec.developpez.com
' Description : Cette routine va permettre
d'ajouter les flèches "suivantes" et
"précédentes"
' =====

' ===== affectation =====
' on récupère le nombre de slides
(diapositives)
i = ActivePresentation.Slides.Count
' pour positionner le haut de nos boutons
intTopBtn =
ActivePresentation.PageSetup.SlideHeight - 100
' dimensions des flèches
intHeightBtn = 50
intWidthBtn = 100

For Each sld In ActivePresentation.Slides
Select Case sld.SlideIndex
Case 1
' on est sur la première diapositive
Call FlecheSuivante
Case i
' on est sur la dernière diapositive
Call FlechePrecedente
Case Else
' on est sur les autres diapositives
Call FlechePrecedente
Call FlecheSuivante
End Select
Next sld
End Sub
```

Création des deux flèches :

```
VBA
Public Sub FlechePrecedente ()
' =====
' Procédure : Module :mod_BoutonForm //
Procédure : FlechePrecedente
' DateTime : 08/03/2008 07:46
' Auteur : Starec - Philippe JOCHMANS -
http://starec.developpez.com
' Description : Cette procédure va permettre
la création de la flèche précédente
' =====
```

```

' ===== variables =====
Dim intLeft As Integer ' position du bouton
par rapport au bord gauche
intLeft =
(ActivePresentation.PageSetup.SlideWidth / 2) -
(intWidthBtn * 1.5)

' ajout du bouton et positionnement

Set shp =
sld.Shapes.AddShape(msoShapeLeftArrow, intLeft,
intTopBtn, intWidthBtn, intHeightBtn)
' action du bouton, et mise en place de la
couleur
With shp
.ActionSettings(ppMouseClick).Action =
ppActionPreviousSlide
.Fill.ForeColor.RGB = RGB(200, 180, 250)
.Name = "Précédente"
End With
' mise en place du texte
With shp.TextFrame.TextRange
.Text = "Précédente"
.Font.Name = "Tahoma"
.Font.Size = 12
.Font.Bold = msoTrue
End With
End Sub

```

```

Public Sub FlecheSuivante()
' =====
' Procédure : Module :mod_BoutonForm //
Procédure : FlecheSuivante
' DateTime : 08/03/2008 07:46
' Auteur : Starec - Philippe JOCHMANS -
http://starec.developpez.com
' Description : Cette procédure va permettre
la création de la flèche Suivante
' =====

' ===== variables =====
Dim intLeft As Integer ' position du bouton
par rapport au bord gauche
intLeft =
(ActivePresentation.PageSetup.SlideWidth / 2) +
(intWidthBtn * 0.5)

' ajout du bouton et positionnement

Set shp =
sld.Shapes.AddShape(msoShapeRightArrow, intLeft,
intTopBtn, intWidthBtn, intHeightBtn)
' action du bouton, et mise en place de la
couleur
With shp
.ActionSettings(ppMouseClick).Action =
ppActionNextSlide
.Fill.ForeColor.RGB = RGB(200, 180, 250)
.Name = "Suivante"
End With
' mise en place du texte
With shp.TextFrame.TextRange
.Text = "Suivante"
.Font.Name = "Tahoma"
.Font.Size = 12
.Font.Bold = msoTrue
End With
End Sub

```

Suppression des flèches :

VBA

```

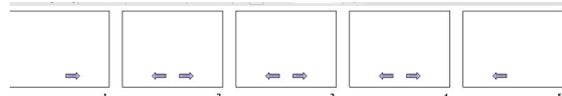
Public Sub SupFleche()
' =====
' Procédure : Module :mod_BoutonForm //
Procédure : SupFleche
' DateTime : 08/03/2008 07:57
' Auteur : Starec - Philippe JOCHMANS -
http://starec.developpez.com
' Description : Suppression des flèches
' =====

' ===== déclaration =====
Dim sld As Slide
Dim shp As Shape
Dim i As Integer
For i = 1 To 2
For Each sld In ActivePresentation.Slides
For Each shp In sld.Shapes
If shp.Name = "Précédente" Or
shp.Name = "Suivante" Then
shp.Delete
End If
Next shp
Next sld
Next i
End Sub

```

Comme vous le voyez, le fonctionnement est similaire, les différences interviennent dans le type de Shape que l'on rajoute, et le texte.

Vous aurez alors cela :



N'hésitez pas à consulter l'explorateur d'objet (F2) et l'aide (F1), vous y trouverez tous les paramètres.

Ces deux touches doivent être les plus usées de votre clavier.

4.3. Cas particulier des images

Le principe avec des images est exactement le même, sauf que lorsque vous voulez ajouter une image, il vous faut utiliser la méthode **AddPicture** de la collection **Shapes**.

5. LES LIENS HYPERTEXTES

Il est possible de naviguer grâce à des liens hypertextes (autres que ceux sur les boutons d'action).

Les liens hypertextes vont vous permettre d'effectuer une action non linéaire, comme on le voit sur le premier schéma de cet article.

Vous en trouverez un exemple et la manière de procéder dans cet article : Les listes à puces et numérotées dans PowerPoint ([Lien 112](#)).

Les liens hypertextes sur la deuxième diapositive (la première servant en général de page d'accueil), couplés avec les méthodes ci-dessus affectées à un bloc de diapositives permettent de réaliser des transitions non linéaires facilement.

6. CONCLUSION

En conclusion, lorsque vous voulez gérer vous-même la navigation dans vos diaporamas, il faut rester logique.

- **Navigation linéaire** : choisissez les boutons d'action qui sont prédéfinis ou les formes.
- **Navigation non linéaire** : vous pouvez réaliser une application non linéaire grâce aux liens hypertextes qui vous permettront de naviguer comme dans une table des matières d'un document Word.

Pensez que les personnes qui utiliseront vos diaporamas ne sont peut-être pas habituées à les manipuler. Le fait de mettre des indications comme des flèches ou autres facilitera grandement l'utilisation de vos diaporamas, de plus chaque lecteur ne lit pas de la même façon et n'a pas la même vitesse de lecture.

Retrouvez l'article de Philippe Jochmans en ligne : [Lien 113](#)

Présentation de Microsoft Office - Excel 2007

Cette page présente l'application Microsoft Office Excel 2007.

1. Introduction

Excel 2007 fait partie de la suite bureautique et base de données intégrée Microsoft Office 2007.

Cet outil puissant permet la création, la mise en forme des feuilles de calcul ainsi que l'analyse et le partage des informations.

Vous pouvez ainsi :

- * stocker, manipuler et mettre en forme les données nécessaires à vos calculs ;
- * trier et filtrer vos données afin de les analyser ;
- * utiliser la mise en forme conditionnelle pour visualiser rapidement certaines informations ;
- * utiliser une large panoplie de fonctions pour la création de formules complexes ;
- * créer des tableaux et graphiques croisés dynamiques ;
- * créer des graphiques pour faire ressortir les principales tendances de vos données ;
- * consulter et manipuler les données dans un navigateur Web ;
- * convertir les feuilles de calcul au format PDF et XPS ;
- * partager et échanger vos résultats grâce aux outils de partage.

Depuis son apparition dans le milieu des années 80, le tableur Microsoft a subi de nombreuses modifications.

Les évolutions n'ont pas toujours été majeures, loin de là, mais les ajouts, améliorations et liftings successifs en ont fait l'outil indispensable pour les calculs, la mise en oeuvre de fonctions complexes et l'analyse de données.

Historique des versions Excel.

- * **1**: 1985, sortie de la version 1 pour Macintosh.
- * **2**: 1987, sortie de la première version pour Windows
- * **3**: 1990.
- * **4**: 1992.
- * **5**: 1993, arrivée des classeurs multifeuilles et du VBA.
- * **7**: 1995, (Excel95) mise en place du Portage 32 bits.
- * **8**: 1997, (Excel97) nouvelle interface VBA, et apparition des UserForms.
- * **9**: 1999, (Excel2000).
- * **10**: 2001, (Excel2002 ou XP), mise en place de l'outil de récupération automatique des classeurs.
- * **11**: 2003, (Excel2003), possibilité de mapper les fichiers XML dans les classeurs.
- * **12**: 2007, (Excel2007) de nombreuses nouveautés et évolutions majeures qui vont être décrites dans cette page.

Cet article propose un simple aperçu des fonctionnalités d'Excel 2007, l'ensemble des nouveautés étant trop

important pour être présenté ici de façon détaillée et exhaustive. D'autres articles suivront pour approfondir les différents thèmes.

2. Généralités

2.1. Le format openXML

Microsoft Excel 2007 introduit un nouveau format de fichier appelé **OpenXML** (XML ouvert Office).

Ce nouveau format :

- * améliore l'interaction et l'utilisation par d'autres applications ;
- * facilite l'intégration aux sources de données externes ;
- * réduit la taille des fichiers (technologie de compression zip utilisée pour stocker les documents) ;
- * améliore la récupération des contenus pour les fichiers endommagés.

L'extension proposée par défaut lors de l'enregistrement d'un classeur est le type **.xlsx**.

Les autres extensions XML sont :

- * classeur autorisant les macros (**.xlsm**) ;
- * modèle par défaut (**.xlt**) ;
- * modèle autorisant les macros (**.xltn**) ;
- * macro complémentaire (**.xlam**) ;
- * le dernier format **.xlsb**, est une version binaire non XML.

La lettre **x** en fin d'extension signifie que le fichier ne contient aucune macro. La lettre **m** signifie que le fichier contient des macros. Si vous enregistrez et fermez votre classeur au format **.xlsx** alors qu'il contenait des macros, celles-ci seront perdues.

Pour plus d'informations, lisez les articles d'Olivier Lebeau et Eric Grimois :

Le XML dans Microsoft Office : [Lien 114](#) ;

Structure des fichiers OpenXML : [Lien 115](#).

Vous pouvez consulter les normes et les conventions de ce standard sur le site ECMA : [Lien 116](#).

2.2. Les limites d'Excel

Ce chapitre reprend simplement quelques points pour mettre en avant l'évolution importante par rapport aux anciennes versions d'Excel.

Pour plus de précisions, utilisez l'aide Office et saisissez le

mot clé **limite**.

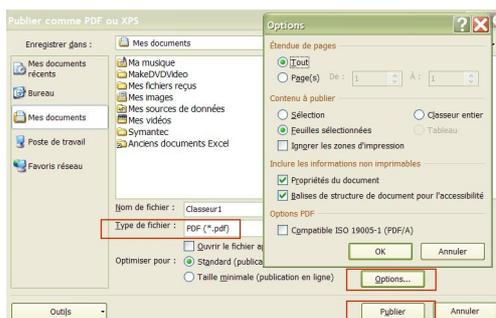
- * Taille des feuilles de calcul : **1 048 576 lignes et 16 384 colonnes**.
- * Nombre maximal de caractères qu'une cellule peut contenir : **32 767**.
- * Nombre maximal de couleurs par classeur : **16 millions de couleurs (32 bits avec accès complet au spectre de couleurs 24 bits)**.
- * Nombre maximal de formats/styles de cellule uniques : **64 000**.
- * niveau de tri : **64 pour un tri simple, illimité lors de l'utilisation de tris séquentiels**.
- * Longueur du contenu des formules : **8 192 caractères**.
- * Niveaux de fonctions imbriqués : **64**.
- * Points de données pour toutes les séries de données d'un graphique : **256 000**.
- * Nombre maximal de points de données par série de données dans les graphiques 2D : **32 000**.
- * Points de données par série de données pour les graphiques 3-D : **4 000**.
- * Nombre maximal d'utilisateurs pouvant ouvrir et partager un classeur partagé en même temps : **256**.

2.3. Enregistrer et exporter aux formats pdf et xps

Il existe un complément pour enregistrer facilement vos fichiers aux formats **PDF** (Portable Document Format) et **XPS** (XML Paper Specification).

Cliquez sur ce lien pour télécharger et activer la prise en charge de ces formats : [Lien 117](#).
Ces types de fichiers seront ensuite disponibles directement dans le sous-menu **Enregistrer sous**.

Des options permettent de sauvegarder le classeur entier, les feuilles sélectionnées, les pages spécifiées ou seulement une plage de cellules.
Cliquez sur le bouton **Publier** pour enregistrer.



2.4. Compatibilité avec les versions antérieures d'Excel

Le menu de vérification est un point important dans Excel 2007 car il liste toutes les fonctionnalités risquant d'être perdues ou dégradées si le classeur est enregistré dans une version d'Excel antérieure.

Par exemple, si vous sauvegardez votre classeur dans une ancienne version d'Excel, les données après la 65536e ligne et de la 256e colonne ne seront pas prises en compte. De la même manière, les formules qui font référence à des cellules au-delà de cette limite poseront aussi un problème.

Une fois les points de différence identifiés, vous pourrez apporter les modifications nécessaires à une meilleure

compatibilité descendante.

Pour lancer la procédure de vérification :
cliquez sur le bouton **Office** ;
sélectionnez le menu **Préparer** ;
cliquez sur le bouton **Activer le vérificateur de compatibilité** ;
une boîte de dialogue va afficher tous les éléments incompatibles, ainsi que des liens d'aide et de correction.

Si vous utilisez une ancienne version d'Excel, vous pouvez lire, modifier et enregistrer les classeurs aux nouveaux formats Microsoft Office 2007 grâce au pack de compatibilité pour Microsoft Office 2000, Office XP ou Office 2003.

Téléchargez le pack de compatibilité : [Lien 118](#).

Comment ouvrir et enregistrer des Excel 2007 dans des versions antérieures d'Office : [Lien 119](#).

2.5. La correction orthographique

Le vérificateur orthographique est commun entre les différentes applications Office. Lorsque vous personnalisez une option de correction dans Excel, la modification sera aussi prise en compte dans Word, Outlook, PowerPoint et dans Publisher.

Le mode français bénéficie de trois options :

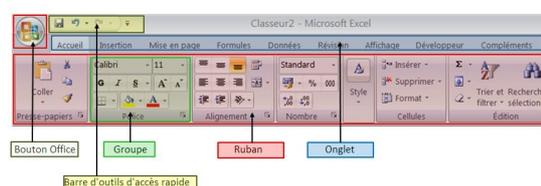
- * Orthographe traditionnelle et rectifiée.
- * Orthographe traditionnelle.
- * Orthographe rectifiée.

Pour accéder à la fenêtre de gestion des corrections :
cliquez sur le bouton **Office** ;
cliquez sur le bouton **Options Excel** ;
sélectionnez le menu **Vérification**.

Ou, depuis l'onglet **Révision** dans le ruban, cliquez sur le bouton **Orthographe** puis sur le bouton **Options**.

3. Le bouton Office et le ruban

Dans Excel 2007, la barre de menus est remplacée par des onglets et un ruban qui rassemble toutes les fonctionnalités disponibles dans l'application.



Cette nouvelle présentation sera peut-être déroutante pour les possesseurs d'anciennes versions d'Excel.

L'adaptation se fait toutefois rapidement. Passez un peu de temps à repérer dans quels onglets sont rangées vos anciennes barres d'outils et familiarisez-vous avec la logique de l'application : les fonctionnalités sont regroupées par thème.

Microsoft met à disposition un support pour faciliter la prise en main du ruban :

Le guide de référence des commandes d'Excel 2003 et

Excel 2007 : [Lien 120](#).

Cliquez sur le lien **Démarrer le guide** pour visualiser la présentation Flash. En sélectionnant les images de la barre d'outils Excel 2003, l'équivalence Excel 2007 s'affiche à l'écran.

Sinon, pour gagner du temps, vous pouvez toujours utiliser les classiques raccourcis clavier afin d'atteindre vos menus.

Enfin, des infobulles détaillées décrivent les fonctionnalités lorsque vous passez le curseur de la souris sur les menus et les éléments de l'application. Cela vous aidera à vous repérer dans ce nouvel environnement.

3.1. Le bouton Office

Le bouton **Office** remplace le menu **Fichier**. Ce bouton est constitué d'un menu et d'une barre d'accès rapide personnalisable.

Il permet aussi d'accéder :

- * aux options Excel ;
- * aux documents récents (les fichiers peuvent être fixés dans la liste grâce à un bouton "**punaise**").

3.1.1. La barre d'accès rapide

La barre d'accès rapide contient trois boutons par défaut (Enregistrer le classeur, Annuler les changements et Rétablir les changements).

Faites un clic droit sur l'objet afin d'accéder à l'option de personnalisation, puis choisissez les boutons qui apparaîtront dans la barre d'outils.

3.1.2. Description des menus

Le sous-menu **Nouveau** permet d'ouvrir un classeur vierge, un modèle personnel ou un des modèles complémentaires disponibles sur le site Microsoft Office Online.

Le sous-menu **Enregistrer sous** propose de sauvegarder le classeur au format de votre choix :

- * au standard **openXML** ;
- * compatible avec les anciennes versions d'Excel ;
- * au format pdf (si le complément a été installé) ;
- * aux autres formats (csv, txt, html, xla, slk...).

Le format présenté par défaut est .xlsx. Cliquez sur le bouton **Autres formats** pour afficher la boîte de dialogue "**Enregistrer sous**".

Le sous-menu **Préparer** contient les outils d'aide à la distribution du classeur :

- * le bouton **Propriétés** pour lire ou modifier vos données personnelles.

Cliquez sur l'option Propriétés du classeurs/Propriétés avancées pour afficher la boîte de dialogue **Propriétés du classeur**.

- * le bouton **Inspecter le document** vérifie la présence de données confidentielles : Les commentaires, les contenus masqués,

et les informations d'identification qui vous permettent de rédiger ou de modifier des documents en

groupe lors du travail collaboratif.

- * le bouton **Chiffrement** protège le classeur par un mot de passe.

- * le bouton **Signature numérique garantit** l'authentification et l'intégrité des classeurs.

- * le bouton **Marquer comme final** permet de spécifier la propriété **Etat** du classeur comme **Final**.

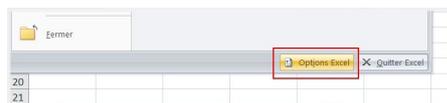
Le fichier est alors uniquement accessible en lecture seule.

- * le bouton **Activer le vérificateur de compatibilité** est une option importante car elle liste toutes les fonctionnalités risquant d'être perdues ou dégradées si le classeur est enregistré dans une version d'Excel antérieure.

Le sous-menu **Publier** gère la distribution et le partage des classeurs.

3.1.3. Les options Excel

Désormais, vous accédez aux options d'Excel à partir du bouton **Office**.



En plus des options classiques, quelques éléments sont à signaler :

Dans le sous-menu **Standard** :

- * vous pouvez personnaliser le jeu de couleurs d'Excel (Bleu, Argenté ou noir) ;
- * cochez l'option **Afficher l'onglet développeur dans le ruban** pour gérer vos projets VBA ;
- * vous pouvez paramétrer les langues à utiliser.

Le sous-menu **Personnaliser** permet de modifier les boutons qui apparaissent dans la barre d'accès rapide.

- * Sélectionnez les commandes dans les différentes listes, puis cliquez sur **Ajouter** ou **Supprimer** afin de personnaliser la barre.

Cette transaction est aussi accessible en faisant un clic droit sur le bouton Office, le ruban ou sur les onglets.

Le sous-menu **Compléments** gère les macros complémentaires.

Le sous-menu **Enregistrement** :

- * vous pouvez spécifier un format de fichier par défaut pour la sauvegarde ;
- * il est possible de paramétrer la récupération automatique des classeurs (option existante depuis Excel2002).

3.2. Le ruban

Le ruban propose des groupes de boutons, arrangés dans neuf onglets.

Certains groupes possèdent une petite icône dans l'angle inférieur droit. Cliquez dessus pour faire apparaître les boîtes de dialogues associées.

Le menu contextuel du ruban permet de paramétrer la barre d'outils **Accès rapide** et de masquer le ruban pour un gain de place à l'écran.

Des onglets supplémentaires spécifiques sont accessibles quand certains objets sont sélectionnés (les images, formes automatiques, SmartArt, graphiques...).

Si vous avez besoin d'aide, profitez des infobulles qui détaillent les fonctionnalités lorsque le curseur de la souris passe sur les menus.

Le passage de la souris sur certains boutons contextuels permet aussi de prévisualiser dynamiquement le résultat qui sera appliqué lors de la validation.

3.2.1. L'onglet Accueil

L'onglet **Accueil** rassemble les commandes générales pour la saisie et mise en forme du tableur.



Le groupe Presse papier :

- * Copier, couper, Coller.
- * Reproduire la mise en forme.
- * Afficher le volet Presse-papiers Office.

Le groupe Police :

- * Le type de police.
- * La taille de la police.
- * Augmenter et diminuer progressivement la taille de la police.
- * Gestion des bordures de cellules.
- * Mise en forme des données (gras, souligné, italique)
- * Couleur de remplissage et la couleur de la police.

Le groupe Alignement :

- * Gestion des alignements verticaux et horizontaux.
- * Orientation du texte.
- * Augmenter ou diminuer le retrait.
- * Renvoyer à la ligne automatiquement.
- * Fusionner les cellules.

Le groupe Nombre :

- * Le format de la cellule
- * Ajout et réduction des décimales.

Le groupe Style :

- * La mise en forme conditionnelle.
- * La mise en forme de tableaux.
- * Les styles de cellules.

Le groupe Cellules :

- * Insérer des cellules, des lignes, des colonnes, des feuilles.
- * Supprimer des cellules, des lignes, des colonnes, des feuilles.
- * Paramétrer la taille des cellules.
- * Masquer des cellules, des lignes, des colonnes, des feuilles.
- * Afficher des cellules, des lignes, des colonnes, des feuilles.
- * Renommer la feuille.
- * Déplacer ou copier la feuille.

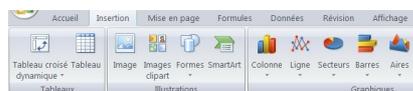
- * Définir une couleur d'onglet.
- * Verrouiller les cellules.
- * Protéger la feuille.
- * Définir le format des cellules.

Le groupe Édition :

- * Accès rapide aux fonctions Somme, Moyenne, Nb, Max, Min.
- * Supprimer les éléments de la cellule.
- * Trier des valeurs ou des couleurs.
- * Filtrer.
- * Rechercher et remplacer dans la feuille.
- * Atteindre.
- * Afficher le volet de sélection (gestion des objets contenus dans la feuille).

3.2.2. L'onglet Insertion

L'onglet Insertion contient les objets pouvant être ajoutés dans une feuille Excel.



Le groupe Tableaux :

- * Insertion de tableaux croisés dynamiques.
- * Insertion de tableau.

Le groupe Illustrations :

- * Insertion d'images et de ClipArt.
- * Insertion de formes.
- * Insertion de SmartArt.

Le groupe Graphiques :

- * En colonne.
- * En ligne.
- * En secteurs.
- * En barres.
- * En aires.
- * En nuage de points.
- * Autres types de graphiques (Surface, Anneau, Bulle, Radar).

Le groupe Liens hypertextes :

- * Insère un lien hypertexte dans la cellule sélectionnée.

Le groupe Texte :

- * Insertion des zones de texte.
- * Insertion des entêtes et pieds de page.
- * Insertion des textes WordArt.
- * Insertion des lignes de signature.
- * Insertion des objets.
- * Insertion de symboles.

3.2.3. L'onglet Mise en page

Cet onglet gère la mise en forme de la feuille de calcul.



Le groupe Thèmes :

- * Liste des thèmes prédéfinis.

Le groupe Mise en page :

- * Orientation (portrait ou paysage).
- * Marges
- * Définition de la taille du papier (A4, enveloppes...)
- * Zone d'impression.
- * Gestion des sauts de page.
- * Gestion des images d'arrière-plan.
- * L'impression des titres (lignes et colonnes à reproduire sur chaque page imprimée).

Le groupe Mise à l'échelle :

Le groupe Options de la feuille de calcul :

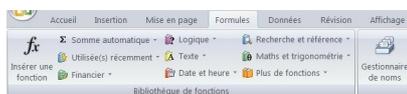
- * Afficher ou masquer le quadrillage.
- * Imprimer le quadrillage.
- * Afficher ou masquer les entêtes de colonnes et de lignes.
- * Imprimer les entêtes de colonnes et de lignes.

Le groupe Organiser :

- * Mise en page des objets contenus dans la feuille.
- * Mettre au premier plan ou à l'arrière-plan.
- * Afficher le volet de sélection (gestion des objets contenus dans la feuille).
- * Aligner les objets.
- * Grouper les objets.
- * Rotation des objets.

3.2.4. L'onglet Formules

L'onglet Formules regroupe toutes les fonctions classées par genre et les options de gestion.



Le groupe Bibliothèque de fonctions :

- * Liste des fonctions regroupées par catégorie.
- * Afficher la boîte de dialogue pour insérer une formule.

Le groupe Noms définis :

- * Gérer les cellules et les plages nommées (ajout, modification, consultation, suppression).

Le groupe Audit de formules :

- * Repérer les antécédents et les dépendants.
- * Supprimer les flèches d'audit.
- * Afficher les formules à la place de leur résultat.
- * Vérifier et repérer les erreurs.
- * Évaluer les formules.
- * Utilisation des fenêtres espions.

Le groupe Calcul :

- * Les modes de calcul (automatique, automatique sauf sur les tables de données, manuel)
- * Calculer le classeur maintenant (F9).
- * Calculer la feuille maintenant (Maj+F9).

Le groupe Solutions :

- * Effectuer des conversions vers et à partir de la devise Euro.

3.2.5. L'onglet Données

L'onglet Données gère les données du classeur et les données externes.



Le groupe Données externes :

- * Création de connexions vers des sources externes (Access, fichiers txt et csv, à partir de sites web).
- * Gestions des connexions existantes.

Le groupe Connexions :

- * Actualisation des informations provenant des sources de données.
- * Propriété des plages de données.
- * Gérer toutes les connexions pour le classeur (ajout, modification, suppression).

Le groupe Trier et filtrer :

- * Trier des valeurs ou des couleurs.
- * Filtrer.

Le groupe Outils de données :

- * Répartir le contenu d'une cellule dans des colonnes distinctes (convertir).
- * Supprimer les doublons.
- * Création des listes de validation.
- * Consolider.
- * Analyser les scénarios.

Le groupe Plan :

- * Gérer les plans (grouper et dissocier).
- * Application d'un sous-total.

Le groupe Analyse :

- * Utilisation de l'utilitaire d'analyse pour les données financières et scientifiques (Funcres).

3.2.6. L'onglet Révision

L'onglet Révision permet la vérification, la gestion des annotations et la protection du classeur.



Le groupe Vérification :

- * Vérification de l'orthographe.
- * Rechercher (dans les dictionnaires, les encyclopédies et les traductions).
- * Dictionnaire des synonymes.
- * Traduction du texte sélectionné.

Le groupe Commentaires :

- * Gestion des commentaires (ajouter, afficher, masquer, supprimer).

Le groupe Modifications :

- * Protéger la feuille.
- * Protéger le classeur.
- * Protéger et partager le classeur.
- * Gérer les plages déverrouillées par un mot de passe

lorsque la feuille est protégée.

- * Suivi des modifications.

3.2.7. L'onglet Affichage

L'onglet Affichage gère la mise en forme de la page à l'écran.



Le groupe Affichages classeur :

- * Afficher en mode normal ou en mode aperçu des sauts de page.
- * Affichage en mode "Mise en page".
- * Afficher en mode plein écran (Nota: la fermeture du mode plein écran est placée dans le menu contextuel de la feuille).

Le groupe Afficher/Masquer :

- * Afficher/Masquer le quadrillage.
- * Afficher/Masquer la barre de formules.
- * Afficher/Masquer les entêtes de lignes et de colonnes.
- * Afficher/Masquer les règles.

Le groupe Zoom :

Le groupe Fenêtre :

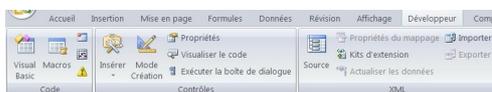
- * Figurer les volets.
- * Ouvrir le classeur dans une deuxième fenêtre.
- * Réorganiser les classeurs (horizontal, vertical, mosaïque, cascade).
- * Fractionner.
- * Masquer la fenêtre.
- * Afficher deux feuilles côte à côte.
- * Synchroniser le défilement de deux classeurs.
- * Enregistrer l'espace de travail.
- * Basculer vers un autre classeur ouvert.

Le groupe Macros :

- * Afficher la liste des macros.
- * Utiliser l'enregistreur de macros.

3.2.8. L'onglet Développeur

Comme son nom l'indique, cet onglet permet de gérer vos macros.



Le groupe Code :

- * Accéder à l'éditeur de macros (Alt+F11).
- * Afficher la classique boîte de dialogue qui liste les macros exécutables (Alt+F8).
- * Utiliser l'enregistreur de macros.
- * Gérer la sécurité des macros.

Le groupe Contrôles :

- * Insérer des contrôles Formulaires ou ActiveX.
- * Activer ou désactiver le mode création.
- * Afficher la fenêtre Propriétés.
- * Visualiser le code de la feuille active.

Nota: Les contrôles Formulaires sont toujours disponibles dans Excel2007 mais il est préférable d'utiliser les contrôles ActiveX qui sont plus simples d'emploi et possèdent plus d'options.

Le groupe XML :

- * Importer et exporter des fichiers xml.
- * Actualiser les données.
- * Mapper dans le classeur.

Si l'onglet **Développeur** n'apparaît pas dans le ruban :

- Cliquez sur le bouton **Office**,
- puis sur le bouton **Options Excel**.
- Cliquez sur le menu **Standard**.
- Cochez l'option **Afficher l'onglet Développeur dans le ruban**.

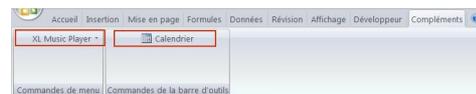
Cliquez sur le bouton **OK** pour valider.

Pour gérer les macros complémentaires :

- Cliquez le bouton **Office**.
- Cliquez sur le bouton **Options Excel**.
- Sélectionnez le menu **Compléments**.
- Choisissez **Compléments Excel** dans le menu déroulant **Gérer** (en bas de la fenêtre).
- Cliquez sur le bouton **Atteindre**.

3.2.9. L'onglet Compléments

L'onglet Compléments donne accès aux macros complémentaires que vous avez activées.



3.3. La barre de formules

La barre de formules est maintenant extensible horizontalement et verticalement.

Il est possible de rédiger des formules plus longues et comportant plus de niveaux d'imbrication qu'auparavant.



Sélectionnez simplement les bords de la barre de formules afin de modifier sa taille.

Le menu déroulant sur la droite (dont la taille est aussi extensible) permet d'accéder aux dernières fonctions utilisées.

4. La feuille de calcul

4.1. Généralités

Pour éviter les risques d'erreur lors de la saisie des formules, Excel 2007 dispose d'un outil de saisie semi-automatique.

Lorsque vous rédigez le début de la formule, l'application propose la liste des fonctions correspondantes. Cette nouveauté est très appréciable. On remarquera que les cellules et plages nommées apparaissent aussi dans la liste (**ProductionAnnee** dans la copie d'écran ci-dessous).



Il est désormais possible de supprimer les doublons facilement:

Sélectionnez votre plage de cellules.

Cliquez sur l'onglet **Données** dans le ruban.

Cliquez sur le bouton **Supprimer les doublons**, dans le groupe **Outils de données**.

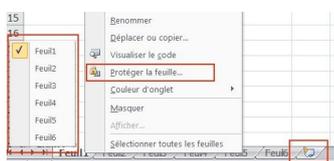
Une option permet de préciser si la première cellule correspond à un entête (elle ne sera pas prise en compte lors de la suppression).

Cliquez sur le bouton **OK** pour valider.



4.2. La barre d'état et la barre d'onglets

La barre d'onglets dispose de quelques fonctionnalités supplémentaires:

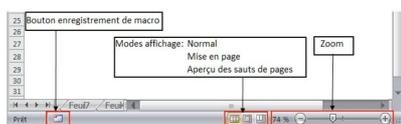


Le menu contextuel contient un accès rapide pour protéger la feuille.

Une feuille peut être ajoutée directement en cliquant sur le dernier onglet, sur la droite de la barre (ou en utilisant le raccourci clavier **Maj+F11**).

Le menu qui liste les feuilles du classeur (clic droit sur la gauche de la barre d'onglets) existait déjà dans les versions précédentes d'Excel, mais il est toujours intéressant de rappeler cette option qui permet d'atteindre rapidement une feuille, surtout quand le classeur en contient beaucoup.

La barre d'état a aussi évolué:



Le menu de Zoom, et notamment son curseur, permet de modifier la zone d'affichage très facilement.

Vous pouvez basculer entre trois modes d'affichage : Normal, Mise en page et Aperçu des sauts de page. Le mode **Mise en page** est très intéressant car vous travaillez sur la feuille de calcul tout en voyant ce que donnera exactement la page lorsqu'elle sera imprimée. Ce mode permet aussi d'accéder aux champs d'entête et de pied de page.

Sur la gauche de la barre se trouve le bouton pour démarrer et arrêter l'enregistreur de macros.

Faites un clic droit dans la barre d'état pour personnaliser

son contenu.

4.3. Le menu contextuel des cellules

Le nouveau menu contextuel (clic droit dans une cellule) s'est enrichi d'une barre de mise en forme.



Cette barre correspond au groupe **Police**, dans l'onglet **Accueil**.

Le menu contextuel dispose aussi de nouvelles options:

- * Filtrer
- * Trier
- * Nommer une plage

4.4. Les tableaux

Excel permet de convertir une plage de cellules en tableau ("Liste" dans Excel 2003).

Les tableaux facilitent le classement et l'exploitation de vos données.

Sélectionnez les cellules puis cliquez sur le bouton **"Tableau"** dans le groupe **Tableaux** de l'onglet **Insertion**. (Ou sur le bouton **"Mettre sous la forme de tableau"** dans le groupe **Style** de l'onglet **Accueil**).

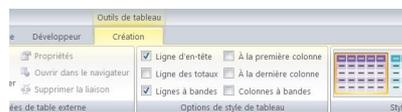
Choisissez un style dans la liste.

La plage est automatiquement mise en forme en fonction du style que vous avez sélectionné.

Un filtre automatique est activé par défaut dans la nouvelle table. Vous pouvez ensuite utiliser la puissance des filtres et des tris pour traiter le contenu des cellules (voir chapitre 4.7).

	A	B	C	D	E
1	Champ01	Champ02	Champ03	Champ04	Champ05
2	Nom04	4 info83	23	30/05/2007	
3	Nom05	5 info84	21	31/05/2007	
4	Nom01	6 info85	51	01/06/2007	
5	Nom01	7 info86	33	02/06/2007	
6	Nom08	8 info87	45	03/06/2007	
7	Nom09	9 info88	72	04/06/2007	

Un nouvel onglet, apparaît dans le ruban lorsque vous sélectionnez n'importe quelle cellule du tableau. L'application comprend que les modifications concernent tout le tableau.



Le groupe Propriétés :

- * Renommer le tableau qui sert de référence (sans espace ni caractère spécial).
- * Redimensionner le tableau.

Le groupe Outils :

- * Synthétiser avec un tableau croisé dynamique.
- * Supprimer les doublons.
- * Convertir en plage (annule la table).

Le groupe Données de table externe :

- * Exporter vers une liste SharePoint ou un diagramme Visio.
- * Actualiser.
- * Spécifier le mode de mise à jour des données.
- * Déconnecter le tableau du serveur.

Le groupe Options de style de tableau :

- * Activer ou désactiver la ligne d'entête.
- * Appliquer une mise en forme pour la première et la dernière colonne.
- * Appliquer des lignes ou des colonnes à bandes (alternance de couleurs).
- * Activer ou désactiver la ligne des totaux.

La ligne de totaux, et les formules qu'elle contient, s'adapte automatiquement lorsque vous redimensionnez le tableau. Vous pouvez définir un type de calcul particulier pour chaque cellule de la ligne de totaux.

Le groupe Styles de tableau :

- * Choisir un autre style dans la galerie.

Les noms de tableau sont accessibles dans le gestionnaire de noms (Onglet Formules/Groupe Noms définis) et aussi visibles dans la liste de saisie semi-automatique des formules.

Vous pouvez donc spécifier le nom du tableau (qui est en fait une plage nommée) dans vos formules :
`=RECHERCHEV("mimi";Tableau1;3;0)`

4.5. La mise en forme conditionnelle

Le nouvel outil de mise en forme conditionnelle dispose d'une large panoplie de règles afin de visualiser et d'analyser les relations ou les tendances entre les données. Les mises en forme conditionnelles sont faciles et rapides à appliquer. Vous pouvez obtenir des synthèses cohérentes et d'une grande qualité visuelle, en quelques clics de souris.

	A	B	C	D	E
1	Champ01	Champ02	Champ03	Champ04	Champ05
2	Nom04	4 info83	23	30/05/2007	
3	Nom05	36 info84	21	31/05/2007	
4	Nom01	6 info85	51	01/06/2007	
5	Nom01	21 info86	33	02/06/2007	
6	Nom08	8 info87	45	03/06/2007	
7	Nom09	38 info88	72	04/06/2007	

Le bouton "Mise en forme conditionnelle" se trouve dans le groupe **Style** de l'onglet **Accueil**.

Après avoir sélectionné une colonne de votre tableau, vous pouvez utiliser les règles prédéfinies dans le menu :

- * Faire ressortir les valeurs (supérieures, comprises entre, contenant, en doublon, inférieures...) par rapport à une variable que vous spécifiez.
- * Faire ressortir les valeurs (les plus élevées et plus faibles en nombre ou en pourcentage, supérieures et inférieures à la moyenne...) par rapport à une variable que vous spécifiez.
- * Appliquer des relations visuelles (nuances de couleurs, icônes indicatrices).
- * Appliquer une barre de données dans chaque cellule, qui correspond à la grandeur de la valeur par rapport à la série.

L'application calcule la mise en forme automatiquement, et s'adapte au contenu des cellules. Les prévisualisations et

les modifications s'effectuent en temps réel. Vous pouvez spécifier plusieurs critères de mise en forme conditionnelle qui s'appliqueront à la plage.

Pour utiliser une formule :

Sélectionnez la ou les cellules.

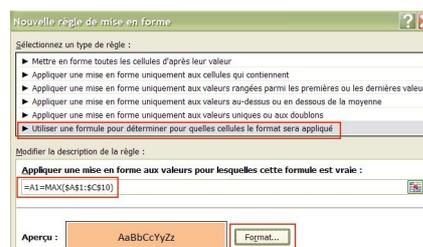
Cliquez sur l'option Nouvelle règle dans le menu du bouton **Mise en forme conditionnelle**.

Sélectionnez **Utiliser une formule...** dans le champ **Type de règle**.

Saisissez votre formule.

Choisissez le format de mise en forme (bouton **Format**).

Cliquez sur le bouton **OK** pour valider.



4.6. La police, les thèmes et les styles

Le type de police

La police de caractères utilisée dans Office 2007 bénéficie de la technologie **ClearType** par défaut. Les polices **ClearType** facilitent la lecture de texte (lissage).

Si dans certaines configurations des caractères ne s'affichent pas ou sont floutés, vous pouvez désactiver ClearType dans les options Excel (menu Standard).

Les thèmes

Les thèmes sont des ensembles prédéfinis de couleurs, de polices, de traits et d'effets de remplissage qui peuvent être appliqués à la feuille de calcul ou à des objets spécifiques. Les thèmes sont communs à toutes les applications Office 2007. La galerie est composée de 20 thèmes par défaut.

Si les modèles proposés ne vous conviennent pas, vous pouvez en télécharger d'autres sur le site Microsoft Office Online :

Sélectionnez sur le bouton **Thèmes** dans l'onglet **Mise en page**.

Cliquez sur le lien **Autres thèmes sur Microsoft Office Online**.

Choisissez le thème qui vous intéresse et suivez les instructions de téléchargement.

Lorsque vous sélectionnez de nouveau le bouton **Thèmes** dans l'onglet **Mise en page**, le thème téléchargé apparaît dans le champ **Personnalisé**.

Les thèmes sont utiles pour standardiser vos présentations et respecter vos chartes graphiques.

Les styles

Si les thèmes sont communs à toutes les applications Office, les styles sont eux conçus pour mettre en forme les éléments spécifiques à Excel

- * Graphiques
- * Tableaux
- * Cellules

- * Tableaux croisés dynamiques
- * SmartArt
- * Formes

Le bouton **Styles de cellules** se trouve dans le groupe **Style** de l'onglet **Accueil**. Un choix important de styles prédéfinis est mis à votre disposition. Vous pouvez facilement créer et personnaliser vos styles si aucun de ceux proposés ne vous convient.

4.7. Les tris et les filtres

Les tris

Vous pouvez facilement classer les données dans la feuille de calcul en utilisant les tris.

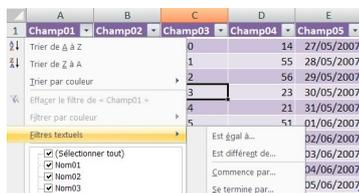
De nouvelles options ont été ajoutées : tri par couleur de cellule ou de police, et jusqu'à 64 niveaux (contre trois dans les versions précédentes).



Les filtres

Profitez aussi des améliorations apportées aux capacités de filtrage :

- * Par couleur.
- * Personnalisé en fonction du type de données (dates, texte, numérique).
- * Augmentation des critères de filtre.



Les options du filtre s'adaptent en fonction du type de données contenu dans la colonne :

- * **Filtres textuels.**
- * **Filtres numériques.**
- * **Filtres chronologiques** pour les dates.

L'application est capable d'analyser automatiquement les dates par jour, semaine, mois, semestre ou année ce qui facilite les filtres sur ce type de données.



Pour dissocier la hiérarchie de dates dans le Filtre (transformation en liste non hiérarchique) :

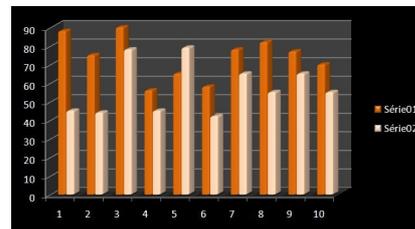
- Cliquez sur le Bouton **Office**.
- Cliquez sur le bouton **Options Excel**.
- Sélectionnez le menu **Options avancées**.
- Dans la partie **Afficher les options pour ce classeur**, Choisissez le classeur.
- Décochez l'option **Grouper les dates dans le menu**

Filtre automatique.

Cliquez sur le bouton **OK** pour valider.

5. Les graphiques

Le nouvel outil de conception des graphiques permet une construction rapide et d'une grande qualité visuelle.



Vous pouvez profiter des nombreuses options disponibles :

- * Choix du type de graphique.
- * Rendu des couleurs et des images.
- * Personnalisation des couleurs par thème ou au choix dans une large palette.
- * Effets visuels (Fonction d'affichage 3D, rotations, ombres adoucies, transparences...).
- * Gestion des modèles et des graphiques par défaut.

Le moteur de création des graphiques est le même dans Office Word et PowerPoint 2007. Vous pouvez donc créer des graphiques de la même manière, dans ces applications.

Le menu de création des graphiques est situé dans l'onglet **Insertion**.



Un nouveau groupe d'onglets est accessible lorsque le graphique est sélectionné :



C'est à partir de ces onglets que vous allez modifier et personnaliser vos graphiques.

Onglet Création :

- * Modifier le type de graphique.
- * Enregistrer comme modèle.
- * Intervertir les lignes et les colonnes.
- * Gérer les plages de données sources.
- * Dispositions du graphique.
- * Sélection du style de graphique depuis une galerie.
- * Déplacer le graphique (dans un onglet graphique ou incorporé dans la feuille de calcul).

Onglet Disposition :

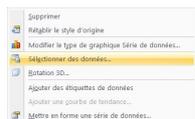
- * Mise en forme de la sélection.
- * Insertion d'images de zones de texte et de formes.
- * Gestion des étiquettes, du titre du graphique, des légendes, des tables de données.
- * Gestion des axes et quadrillages.
- * Gestion des planchers et parois, fonctions d'affichage 3D et des effets visuels.
- * Mise en œuvre des courbes de tendance, barres d'erreur.

- * Propriétés pour renommer le graphique.

Onglet Mise en forme :

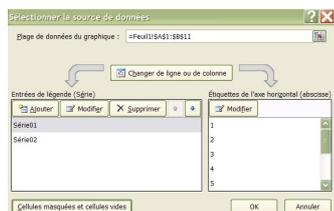
- * Styles de formes, contours, remplissages, effets.
- * Styles WordArt pour les étiquettes, le titre du graphique et les légendes.
- * Gestion des graphiques : liste des graphiques dans un volet de sélection, avec possibilité de changer les titres et masquer/réafficher les graphiques.
- * Modification de la taille du graphique.

Vous pouvez aussi accéder aux options du graphique à partir du menu contextuel (clic droit dans le graphique). Un menu contextuel différent s'affiche en fonction de la zone sélectionnée.



L'option **Sélectionner des données** (aussi accessible par le groupe Outils de graphique/Création/Données/Sélectionner des données) permet d'afficher la boîte de dialogue pour gérer les sources de données.

- * Définir la plage de données.
- * Ajouter, modifier, déplacer ou supprimer des séries.
- * Intervertir les lignes et les colonnes.
- * Modifier les étiquettes de l'axe des abscisses.
- * Gérer les cellules vides et les cellules masquées.

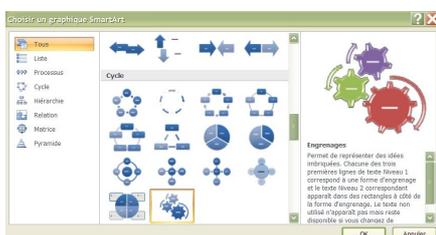


6. Les SmartArt

Excel 2007 intègre la fonctionnalité **SmartArt** qui propose une vaste galerie de pyramides, de diagrammes et d'organigrammes, à placer dans les feuilles de calcul. Cet nouvel outil permet l'illustration et la mise en valeur de vos résultats.

Cliquez sur l'onglet Insertion. L'icône SmartArt est située dans le groupe Illustration. Les diagrammes sont classés par catégorie :

- * Tous
- * Liste
- * Processus
- * Cycle
- * Hiérarchie
- * Relation
- * Matrice
- * Pyramide

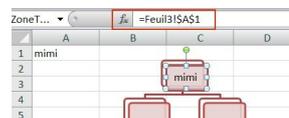


Deux nouveaux onglets (Création et Format) sont accessibles lorsqu'un diagramme SmartArt est sélectionné. Ceux-ci contiennent une large palette d'outils pour la création et la mise en forme des objets.



Il est possible d'afficher le résultat d'une cellule dans un SmartArt :

- * Insérez un diagramme SmartArt dans la feuille de calcul.
- * Appliquez votre mise en forme et finalisez l'aspect visuel.
- * Sélectionnez ensuite la bordure encadrant l'ensemble des formes.
- * Utilisez le raccourci clavier Ctrl+A.
- * Toutes les zones sont ainsi sélectionnées.
- * Effectuez un **Copier**.
- * Sélectionnez une cellule dans la feuille de calcul.
- * **Collez**.
- * Vous obtenez une copie du SmartArt, mais de type forme automatique classique.
- * Vous pouvez à ce stade supprimer le "modèle" initial si vous le souhaitez.
- * Sélectionnez la zone de texte dans les différentes formes et spécifiez la référence à une cellule, dans la barre de formule.
- * Validez la formule.
- * Désormais, le résultat de la cellule Feuil3!\$A\$1 s'affiche dans la forme.

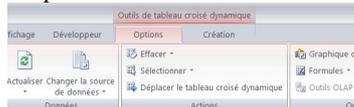


7. Les tableaux et graphiques croisés dynamiques

Les tableaux croisés dynamiques (TCD) servent à synthétiser et analyser des lignes d'information à partir des paramètres que vous avez spécifiés. Comme les autres outils de l'application Excel, les tableaux croisés dynamiques bénéficient aussi d'une nouvelle interface.

Vous pouvez créer un tableau croisé dynamique à partir de l'onglet **Insertion**.

Deux onglets spécifiques (Options et Création) sont accessibles lorsqu'une cellule du TCD est sélectionnée.



L'onglet Options

- Le groupe Options du tableau croisé dynamique :**
- * Modifier le nom du TCD.
 - * Afficher la boîte de dialogue **Options du TCD**.

- Le groupe Champ actif :**
- * Modifier le nom du champ.
 - * Développer ou réduire tous les éléments du champ actif.
 - * Afficher la boîte de dialogue **Paramètres de champs**.

Le groupe Groupe :

- * Créer un groupe contenant les éléments sélectionnés.
- * Dissocier le groupe.
- * Grouper les champs numériques ou date.

Le groupe Trier :

Le groupe Données :

- * Actualiser le TCD.
- * Changer la source de données.

Le groupe Actions :

- * Supprimer les champs, les mises en forme et les filtres.
- * Sélectionner un élément du TCD (étiquettes, valeurs, TCD complet).
- * Déplacer le TCD dans le classeur.

Le groupe Outils :

- * Créer un graphique croisé dynamique.
- * Créer et modifier les champs et éléments calculés.
- * Lister les formules.
- * Travailler sur TCD connecté à une source de données OLAP.

Le groupe Masquer/Afficher :

- * Afficher ou masquer la liste des champs.
- * Afficher ou masquer les boutons +/-.
- * Afficher ou masquer les entêtes de champ des lignes et colonnes.

L'onglet Création

Le groupe Disposition :

- * Paramétrer l'affichage et la disposition des sous-totaux.
- * Activer ou désactiver les lignes et les colonnes.
- * Ajuster la disposition du TCD (compactée, mode plan, tabulaire).
- * Ajouter ou supprimer des sauts de ligne après chaque élément.

Le groupe Options de style de tableau croisé dynamique :

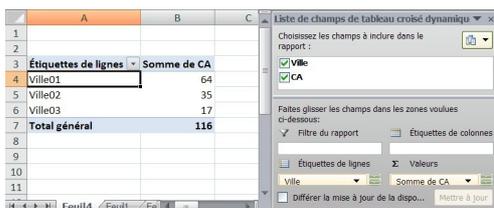
- * Afficher ou masquer la mise en forme des lignes et colonnes.

Le groupe Styles de tableau croisé dynamique :

- * Choisir un style dans la galerie.

Le contenu du TCD est géré à partir du volet **Liste de champs de tableau croisé dynamique**.

Sélectionnez les champs à inclure dans le rapport.



Vous pouvez ensuite utiliser les fonctions pour synthétiser,

analyser et mettre en forme vos données :

- * Appliquer une mise en forme conditionnelle.
- * Trier la colonne de la cellule sélectionnée.
- * Annuler les actions effectuées pour changer l'orientation de l'analyse.

Consultez le tutoriel qui décrit en détail les tableaux croisés dynamiques dans Excel 2007 : [Lien 121](#).

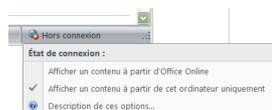
8. L'aide

L'accès à la fenêtre d'aide Office est représenté par une icône bleue, de forme ronde et contenant un point d'interrogation. L'aide est aussi toujours accessible par le raccourci clavier **F1**.

De nombreux exemples très détaillés sont disponibles, permettant de répondre à vos questions et aussi de découvrir toutes les fonctions de l'application.

Une partie de l'aide est stockée sur le poste de travail. Certaines rubriques complémentaires sont hébergées en ligne, sur le site Office Online. En activant la connexion, vous avez accès à un plus grand nombre de procédures d'aide. On peut imaginer le formidable potentiel que cela représente si cette base de connaissance en ligne est régulièrement actualisée et complétée.

Le bouton de connexion au site Office Online est placé dans le coin inférieur droit de la fenêtre d'aide. Vous pouvez paramétrer une utilisation de l'aide, en local ou en ligne, depuis le menu



9. Conclusion

Les nouvelles fonctionnalités sont multiples dans cette dernière version d'Excel. Cet article a à peine effleuré les possibilités offertes par l'application.

Toutes les astuces et les subtilités restent à découvrir au fil de vos projets.

Cette version est une étape importante dans l'évolution de l'application Excel, axée sur l'analyse des données, la mise en forme visuelle et le partage des résultats.

Retrouvez l'article de SilkyRoad en ligne : [Lien 122](#)

Créer des requêtes simples

Cet article est le premier d'une série dédiée aux débutants. Ce premier volet se propose de présenter les différents types de requêtes sous Access et de voir comment on peut :

- créer des requêtes simples ;
- trier les données dans la structure de la requête ;
- imprimer les résultats de la requête.

1. INTRODUCTION

Souvent à la source des formulaires, des listes déroulantes, des zones de listes, des états, la requête est un outil indispensable à connaître.

Je vais donc, dans cette série d'articles, montrer comment générer des requêtes et les manipuler.

Voici les titres des différents articles qui composent cette série :

- 1) Créer des requêtes simples ;
- 2) Utiliser des opérateurs dans les champs calculés et dans les critères ;
- 3) Manipuler les jointures ;
- 4) Créer des requêtes paramétrées ;
- 5) Manipuler les différents types de requêtes.

Je ne parlerai pas des assistants de création de requêtes, l'objectif étant de comprendre comment s'articule une requête et comment on la manipule.

2. VOCABULAIRE

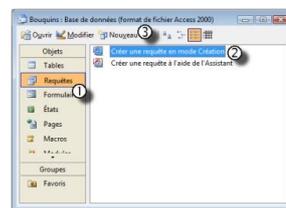
Une requête est une instruction **SQL** permettant de sélectionner un ensemble de données, répondant à un ou plusieurs **critères**, issues d'une ou plusieurs **tables**. La requête peut afficher tout ou partie de l'ensemble des **champs** dans une structure totalement différente de la **structure de l'enregistrement** des tables "source" et en y appliquant une règle de **tri** (contenant de une à dix **clés**).

La requête est créée par l'intermédiaire d'une interface conviviale que l'on appelle **QBE**

Expression	Définition
Base de données	Ensemble d'informations concernant une population organisées en tables.
Table	Ensemble des données organisées en enregistrements. Chaque table stocke les données se rapportant à un objet. Lorsqu'une information présente une redondance, on crée une nouvelle table. Pour plus d'informations sur la conception des tables lire l'article de : Maxence Hubiche : Access - Les bases (Lien 123)

Enregistrements	Ensemble des données organisées suivant une structure concernant un individu de la population.
Structure de l'enregistrement	Ensemble des champs décrivant un individu de la population
Champ	Information élémentaire générique concernant un individu de la population. Exemple : le champ "NomEmploye" ou encore le champ "CodePostal". Remarque : On ne stocke qu'une seule information par champ.
SQL	Structured Query Language : langage d'interrogation des bases de données. Pour plus d'informations sur le langage SQL je vous conseille de lire l'article de SQLPro : Le SQL de A à Z - Les fondements (Lien 124)
Critère	Valeur de sélection appliquée à un champ. Exemple : CodePostal >= 45000
Clé	Nom du champ sur lequel sera exécuté le tri. Exemple : Trier sur le CodePostal par ordre croissant
QBE	Sigle de l'expression Query by Example Interface permettant à l'utilisateur de créer ses requêtes en manipulant directement les champs dans une structure. La syntaxe SQL étant générée automatiquement par Access en arrière-plan.

3. L'ENVIRONNEMENT



Pour créer une nouvelle requête (versions antérieures à Access 2007) :

- cliquez sur l'onglet "Requêtes" (1) puis
- cliquez soit sur "Créer une requête en mode Création" (2)
- soit sur "Nouveau" (3) (Access ouvre une boîte de dialogue donnant accès alors à plusieurs assistants).

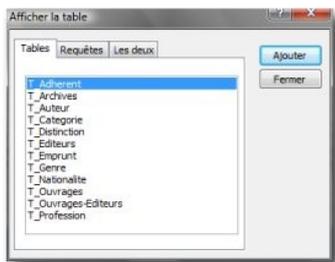


Pour créer une nouvelle requête (version 2007) :

- cliquez dans le ruban sur l'onglet "Créer" (1)
- cliquez sur l'outil "Création de requête" (2)

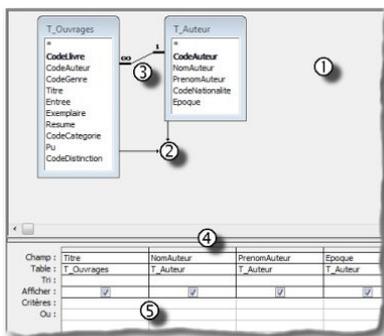
3.1. L'écran QBE

QBE (Query by example) : interface graphique facilitant la création des requêtes et la génération de la syntaxe SQL



Lors de l'appel à la création de la nouvelle requête, Access nous demande de sélectionner une ou plusieurs tables (voire même une ou plusieurs requêtes).

3.1.1. Les zones de travail



Nous allons identifier les différents éléments avec lesquels nous travaillerons lors de la création d'une requête.

Commençons par recréer l'écran ci-contre à gauche. (Téléchargez et ouvrez la base exemple : **Bouquins**)

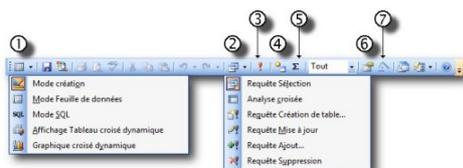
1. Cliquez sur le bouton **Requêtes**.
2. Cliquez sur "**Créer une requête en mode création**".
3. Sélectionnez les tables **T_Auteurs** et **T_Ouvrages**.
4. Cliquez sur le bouton **Ajouter**.
5. En double-cliquant sur les noms de champs des tables situées dans la partie haute de l'écran, ceux-ci apparaissent dans la structure de la requête.

Nous pouvons maintenant passer à l'identification de notre

écran de travail :

1. **Volet des tables** : contient toutes les tables sources de la requête ;
2. **Tables "source"** : contiennent les champs disponibles (à noter la clé primaire affichée en gras) ;
3. **La jointure** : c'est la relation qui existe entre les deux tables. (Ce sujet sera vu en détail dans un article spécifique) ;
4. **La structure de la requête** : regroupe tous les champs nécessaires à l'extraction des informations souhaitées ;
5. **La zone de critères** : partie de l'écran dans lequel nous pourrions combiner les différentes valeurs de sélection.

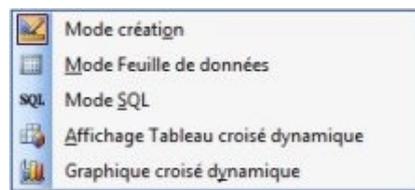
3.1.2. La barre d'outils



Nous allons identifier les outils avec lesquels nous allons principalement travailler.

1. Permet de changer le mode d'affichage et de passer du mode création au mode feuille de données (entre autres) ;
2. Permet de changer le type de la requête en cours de création ;
3. Permet d'exécuter la requête (Je reviendrai sur la différence entre cet outil et l'outil (1) correspondant à l'affichage en feuille de données ;
4. Permet d'ajouter une table à la requête en cours ;
5. Permet d'ajouter la ligne "Regroupement" afin de générer des calculs statistiques dans la requête ;
6. Affiche la fenêtre des propriétés de la requête ou du champ sélectionné ;
7. Ouvre la fenêtre du générateur d'expression facilitant la création d'expression calculée.

En cliquant sur la partie liste déroulante des boutons (1) et (2), Access ouvre des sous-menus découvrant d'autres outils.

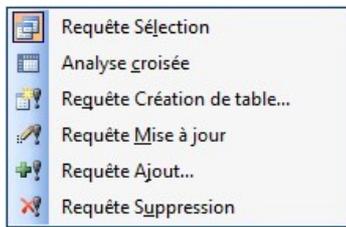


Dans le sous-menu "Affichage", nous trouvons dans l'ordre :

- Affiche le **mode création** c'est-à-dire **la structure de la requête** ;
- Affiche le résultat de la requête sélection sous forme d'une feuille de données (comme dans Excel) ;
- (À remarquer que ce bouton affichera les enregistrements concernés s'il s'agit d'une requête Action (Mise à jour, Suppression, Ajout).)
- Affiche la syntaxe SQL générée par Access (très pratique pour la récupération et l'implantation dans le code

VBA) ;

- Affiche un Tableau Croisé en rapport avec les données récupérées par la requête ;
- Affiche un Graphique Croisé en rapport avec les données récupérées par la requête.



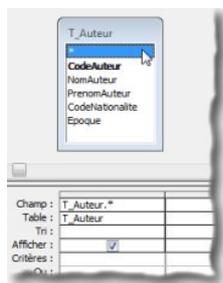
Dans le sous-menu "Type de requête", nous trouvons dans l'ordre :

- **Requête Sélection** : il s'agit de la requête de base permettant d'extraire des enregistrements en fonction d'un ou plusieurs critères ;
- **Requête Analyse croisée** : requête statistique sur deux ou plusieurs niveaux de regroupements ;
- **Requête Création de table** : comme son nom l'indique, permettra de transformer l'extraction obtenue en données d'une nouvelle table ;
- **Requête Mise à jour** : permettra de mettre à jour l'ensemble des enregistrements d'une table en fonction d'un ou plusieurs critères ;
- **Requête Ajout** : permettra de récupérer des données d'une table et de les ajouter dans une autre table de la base en cours ou dans une autre base de données ;
- **Requête Suppression** : permettra de supprimer tous les enregistrements d'une table répondant aux critères posés.

3.2. Création de la structure de la requête

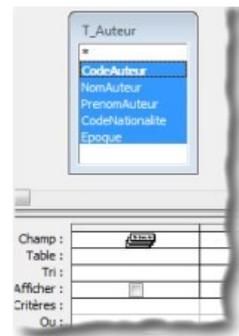
La structure de la requête est composée des différents champs sélectionnés dans les tables "source". Voyons ci-dessous les différentes méthodes permettant d'insérer ou d'ajouter des champs à la structure.

Utiliser le Caractère * :



Double-clic sur l'étoile en haut des champs, L'étoile vient s'afficher dans le premier champ dans la zone de structure. Cette étoile représente l'ensemble des champs de la requête

Sélectionner tous les champs :

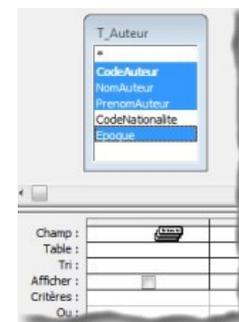


Double-clic sur le nom de la table.

L'ensemble des champs est sélectionné.

Clic maintenu sur la zone sélectionnée et glissez dans la zone de structure de la requête.

Sélections multiples :



Sélectionnez les champs comme on sélectionne les fichiers dans l'explorateur de fichiers. (Utilisation des touches SHIFT et CTRL)

On remarquera que si un champ est déjà sélectionné, on pourra le désélectionner en maintenant CTRL et en re cliquant sur celui-ci.

Insérer un Champ :



Clic maintenu sur le champ à insérer dans la structure et glissez sur le nom du champ qui doit se décaler.

Ajouter un champ :

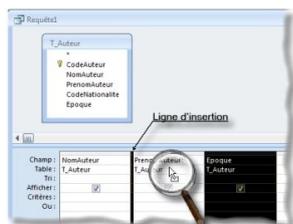
Il suffit de double-cliquer sur le nom du champ à ajouter. Celui-ci vient s'implanter après le dernier champ de la structure.

Déplacer un champ - Étape 1 :



Cliquez sur le sélecteur de champ, petite barre grise, placée au-dessus du champ concerné.

Déplacer un champ - Étape 2 :



Sans bouger la souris, cliquez à nouveau sur la sélection (le pointeur change de forme (cf. la loupe sur l'image)) et glissez pour amener la ligne d'insertion à l'endroit souhaité puis lâchez la souris.

4. CRÉATION DE REQUÊTES SIMPLES

Dans ce chapitre nous allons créer quelques exemples de requêtes simples.

- Requête monotable ;
- Requête multitables avec tri.

4.1. Requête monotable

Il s'agit en fait d'une requête sur une seule table. Couramment utilisée comme source pour les listes déroulantes, elle permet de ne sélectionner qu'une partie des champs de la table en changeant l'ordre de ceux-ci ou en appliquant des règles de tris.

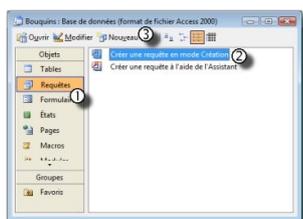
Avant de créer la structure de la requête, définissons notre objectif :

Nous voudrions extraire de la table T_Auteurs, le Nom et le Prénom de celui-ci mais seulement ceux ayant vécu au 19^e siècle.

4.1.1. Création de la requête

Suivant la version, procédez comme suit :

Versions antérieures à 2007 :



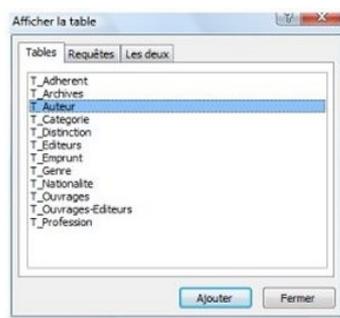
1. Cliquez sur l'onglet **Requêtes**.
2. Cliquez sur **Créer une requête en mode création**.
3. ou cliquez sur l'outil **Nouveau**.

Version 2007 :

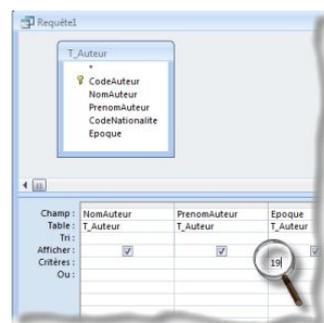


1. Cliquez dans le ruban sur l'onglet **Créer**.
2. Cliquez sur l'outil : **Création de requête**.

Access ouvre l'écran QBE et une boîte de dialogue affichant l'ensemble des tables disponibles.



1. Double-cliquez sur la table **T_Auteur**.
2. Fermez la boîte de dialogue des tables.



1. Double-cliquez sur les différents champs devant faire partie de la requête.
2. Indiquez dans la zone critères, sous le champ concerné, la valeur du **critère** de sélection.

4.1.2. Enregistrement de la requête

Quelle que soit la version d'Access, la génération d'une requête engendre l'ajout d'un nouvel objet qui pourra être mémorisé pour une réutilisation. Ainsi, lorsque notre nouvelle requête est mise au point, il suffit de fermer celle-ci pour qu'Access nous propose d'enregistrer notre travail. Le nom de la requête sera limité à 64 caractères (espaces compris) et sera toujours précédé du préfixe "R_" ou "req" ou "qry".

Enregistrer votre première requête sous le nom : **R_AuteursDu19eme**.

Deux remarques sur l'enregistrement de la requête.

1. Le fait de donner un préfixe permet par la suite de retrouver par ordre les tables, les requêtes et les états.
On sait ainsi toujours quel type d'objet on manipule.
2. Remarquez que dans le nom, j'ai fait abstraction des espaces mais pour un confort de lecture j'ai mis des majuscules à chaque mot.

Cela me facilitera la tâche plus tard lorsqu'il faudra faire

appel au nom d'objet dans les macros ou dans la programmation.

Je serai ainsi dispensé de frapper les [] pour encadrer le nom.

4.1.3. La syntaxe SQL

Quelle que soit la version, Access génère en arrière-plan le code SQL qui sera enregistré lors de la phase précédente. L'objectif, pour nous, ne sera pas de manipuler celui-ci mais de :

1. savoir le visualiser ;
2. savoir le récupérer pour le coller dans un Post lors d'une demande de soutien dans un forum tel que celui-ci : [Lien 125](#)

4.1.3.1. Retrouver le code SQL



Passez en mode création de la requête. En cliquant le bouton droit de la souris dans la zone de tables, choisissez dans le menu contextuel **Mode SQL**.

L'image ci-dessus nous montre l'instruction qui sera enregistrée. J'y ai mis trois repères :

1. représente la clause **SELECT**.
Ce sont les champs sélectionnés qui seront affichés dans le résultat de la requête ;
2. représente la clause **FROM**.
Contient le nom de la ou des tables "source" ;
3. représente la clause **WHERE**.
Partie contenant les critères de sélection.

4.1.3.2. Récupérer le code pour le coller

Sélectionnez par un cliquer-glisser l'ensemble du texte affiché dans la fenêtre.

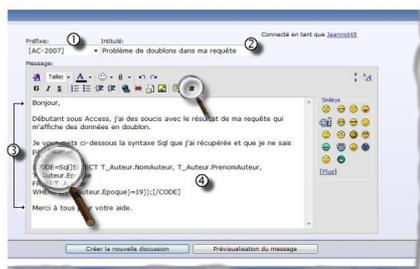
Appliquez le raccourci : **Ctrl C** (correspond à la commande **Copier**).

À partir de maintenant, cette récupération du texte peut être replacée dans n'importe quelle application.

Voyons comment la placer dans un post en cours de composition sur DVP. Connectez-vous sur le Forum ACCESS de DVP et à condition d'être inscrit, vous pouvez ouvrir votre discussion.

Dans un premier temps, vous choisissez le bon sous-forum (en l'occurrence ici : [Lien 126](#)).

Cliquez sur le bouton : **Nouvelle discussion**.



Vous découvrez l'image ci-dessus sur laquelle je ferai quatre remarques :

1. choisissez dans la liste déroulante la version d'Access avec laquelle vous travaillez ;

2. saisissez un intitulé pour la discussion. Que celui-ci soit clair et explicite ;

3. commencez et finissez toujours par un petit mot gentil ;

4. dans le corps de votre texte, cliquez sur le bouton # (marqué par la loupe en haut dans la barre d'outils).

Deux balises [CODE]/[/CODE] apparaissent. Faites alors **CTRL V** (correspond à la commande **Coller**).

Votre syntaxe s'affiche. Vous pouvez ajouter dans la première balise la mention **=SQL** (voir la seconde loupe sur l'image).

Relisez votre message et validez-le.

4.1.3.3. Coller du code SQL dans une requête

Après avoir demandé de l'aide ou des explications sur votre forum préféré, une réponse vous aura été très vraisemblablement donnée.

Une version corrigée de votre code pourra même faire partie de la réponse. Afin d'éviter toute erreur de transcription, je vous conseille de **Copier/Coller** la solution proposée.

Voici ci-dessous les différentes étapes de la manipulation.



1. Sélectionnez la partie de code qui vous intéresse.
2. Appliquez le raccourci **Ctrl C** (correspond à la commande **Copier**).
3. Revenez dans votre fenêtre Access.
4. Créez **une nouvelle requête** (cela préservera votre requête originale).
5. Au moment de sélectionner les tables, fermez la boîte de dialogue (l'interface QBE est affichée sans table).
6. Cliquez le bouton droit de la souris dans la zone de tables et choisissez **Mode SQL** dans le menu contextuel (une fenêtre s'affiche avec le mot **SELECT** sélectionné).
7. Appliquez le raccourci **Ctrl V** (correspond à la commande **Coller**) ; (vous retrouvez le code issu de la réponse du Forum).
8. Réaffichez votre requête en mode Création.
Vous retrouvez votre environnement de travail habituel dans l'interface QBE c'est-à-dire :

- Tables ;
- Jointures ;
- Champs ;
- Critères.

4.2. Requête multitable

On peut remarquer dans notre modèle relationnel que certaines tables contiennent des informations complémentaires.

Exemples :

La table **T_Auteurs** contient une information dont le détail se trouve dans la table **T_Nationalité**.

La table **T_Emprunts** contient des informations dont le détail se trouve dans la table **T_Livres** et dans **T_Adherents**.

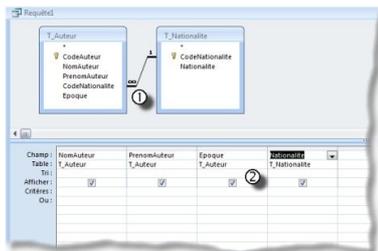
La requête multitable va nous permettre de recréer une information complète en récupérant les données de chacune des tables.

Avant de créer la structure de la requête, définissons notre objectif.

Nous voudrions obtenir la liste de nos auteurs avec le libellé de leur nationalité. Cette requête affichera :

- le prénom de l'auteur ;
- le nom de l'auteur ;
- l'époque ;
- la nationalité.

4.2.1. Création de la requête



Comme précédemment, créez une nouvelle requête et affichez les différentes tables qui seront nécessaires pour la création du travail demandé.

Double-cliquez sur tous les champs nécessaires (cf. image ci-contre).

1. Access affiche la jointure entre les deux tables et le type de relation qui les lie.
2. Dans la ligne des tables, sous les champs, on retrouve les tables sources des champs sélectionnés.

4.2.2. Remarques sur la requête

Je voudrais attirer votre attention sur la "liaison" entre les deux tables sélectionnées pour la requête. On pourra distinguer trois cas.

1) **La relation est établie dans la fenêtre du Modèle relationnel** en paramétrant l'intégrité référentielle qui est un garant de la cohérence des données.

Ce qui signifie que :

1. tout enregistrement Fils doit avoir un enregistrement Père ;
2. il est impossible de supprimer un enregistrement Père sans supprimer le ou les enregistrements Fils attachés.

On retrouvera alors la relation affichée avec les symboles indiquant les cardinalités de la relation.

2) **La jointure sera automatique** à condition que dans les deux tables :

1. il y a un champ identique ayant le même nom ;
2. le même type ;
3. la même taille ;
4. l'un des deux est une clé primaire.

3) **Access ne gère pas automatiquement la jointure.**

La jointure se fera manuellement. Dans ce cas, il faudra cliquer maintenu sur le premier champ de la jointure de la première table et le glisser sur le champ correspondant dans la seconde table. Access ne vérifie rien dans ce cas (on pourrait ainsi joindre un champ numérique avec un champ de type Texte).

Nous reviendrons plus en détail sur la manipulation des jointures dans les requêtes dans un autre article. Pour

patienter, je vous encourage à vous plonger dans le tuto de : **Maxence Hubiche** : [Lien 127](#).

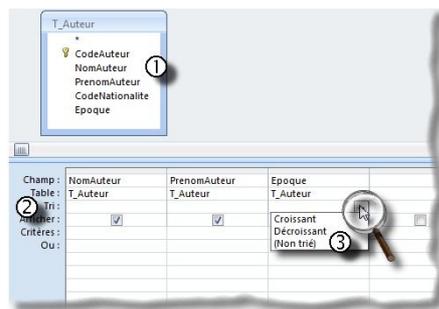
5. TRIER LES DONNÉES D'UNE REQUÊTE

Cette partie de l'article s'attachera à montrer comment générer des tris à partir de la structure de la requête :

1. Tri monoclé ;
2. Tri multiclé.

5.1. Tri monoclé

Afin d'illustrer la méthode, nous allons créer une requête qui nous affichera le Nom et le Prénom de l'auteur trié par **ORDRE CROISSANT** sur le champ **Epoque**.



Pour générer un tri sur une seule clé dans une requête, il suffit :

1. d'implanter les champs nécessaires ;
2. dans la ligne **tri**, cliquer sous le champ clé ;
3. dans la liste déroulante, choisir l'ordre de tri à appliquer

On peut aussi changer l'ordre de tri en double cliquant dans la ligne tri sous le champ concerné

5.2. Tri multiclé

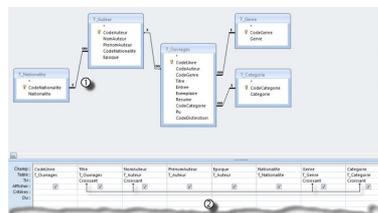
Comme précédemment, l'exemple nous permettra de bien saisir la manipulation.

On se propose donc de créer un catalogue pour notre bibliothèque. Il nous faut faire apparaître dans notre structure les champs suivants :

- le CodeLivre (table : T_Ouvrages) ;
- le Titre du livre (table : T_Ouvrages) ;
- le Nom de l'auteur (table : T_Auteur) ;
- le Prénom de l'auteur (table : T_Auteur) ;
- l'Epoque (table : T_Auteur) ;
- la Nationalité de l'auteur (table : T_Nationalite) ;
- le Genre du livre (table : T_Genre) ;
- la Catégorie de lecteur (table : T_Categorie).

Ce Catalogue sera trié par Catégorie, Genre, Nom de l'auteur, Titre.

Cette requête sera enregistrée sous le nom : **R_CatalogueGeneral**.



Dans l'image ci-dessus, nous retrouvons les différents champs que nous souhaitons intégrer dans notre requête.

1. Nous remarquerons au passage les jointures qui nous indiquent qu'une intégrité référentielle a été appliquée lors de la création des relations.

(Voir ci-dessus : Les remarques sur la requête (Lien 128))

2. Nous affectons les clés de tri comme indiqué précédemment.

Voyons ci-dessous le résultat obtenu :

Bien que nous ayons appliqué les contraintes de tri, il semblerait que le résultat ne soit pas très probant.

En effet, la première règle imposée était : **Trier par catégorie**. En observant le résultat, la requête retourne les enregistrements triés sur le titre.

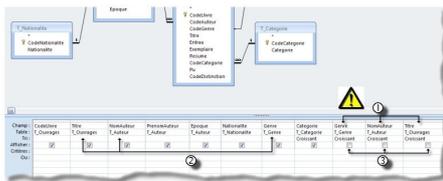
En fait, Access lit les clés de tri de la gauche vers la droite. Ainsi les clés sont appliquées dans l'ordre suivant :

1. Titre ;
2. Nom de l'auteur ;
3. Genre ;
4. Catégorie .

Or nous avons une organisation à respecter : **Ce Catalogue sera trié par Catégorie, Genre, Nom de l'auteur, Titre**

Il faut donc mettre les champs dans l'ordre des clés que l'on souhaite appliquer mais cela va bouleverser l'ordre des champs dans la requête. Pour éviter ce souci, nous allons redessiner notre requête.

Voyons ci-dessous la nouvelle organisation :



Je ferai trois remarques à propos de la structure de la requête ci-dessus.

1. Les champs correspondant aux clés de tri ont été doublés à la fin de la structure de la requête en respectant l'ordre hiérarchique souhaité ;

2. La mention **Croissant** a été retirée sous les trois champs précédents ;

3. La mention **Croissant** a été ajoutée sous les nouveaux champs insérés et j'ai désactivé l'option **Affichage**.

Voyons le résultat obtenu dans l'image ci-dessous :

Ainsi pour trier suivant un ordre particulier tout en préservant une structure de requête, il suffira de répéter les champs clés dans l'ordre hiérarchique souhaité et de désactiver l'affichage de ceux-ci.

6. IMPRIMER LE RÉSULTAT DE LA REQUÊTE

6.1. Un peu de mise en forme

Attention !!!

Quelle que soit la version, toute application de mise en forme s'appliquera à l'ensemble de la feuille de données.

Néanmoins, un petit progrès avec Access 2007, nous pourrons faire une mise en forme par requête.

6.1.1. Versions antérieures à 2007

Boîte de dialogue :



Résultat obtenu :

Après avoir exécuté la requête, cliquez dans le menu **FORMAT** puis sur la commande **Feuille de données...**

Dans la boîte de dialogue, cliquez dans l'option correspondant au type de bordure souhaitée.

6.1.2. Version Access 2007

Dans l'onglet **Accueil** on découvre deux outils qui vont nous permettre de mettre en forme rapidement le résultat afin de faire une impression correcte sans devoir se lancer dans la conception d'un état.

Menu :



Résultat obtenu :

Code Livre	Titre de l'ouvrage	Nom/Auteur	Prénom/Aut.	Epoque	Nationalité	Genre	Catégorie
30	20 000 lieues sous les mers	VERNE	Jules	19	Française	Aventure	Adolescent
30	Trois et les oranges bleues	VERNE	Jules	19	Belge	Bande dessinée	Adolescent
14	Caroline	VOITARE		18	Française	Classique	Adolescent
18	Zelig	VOITARE		18	Française	Classique	Adolescent
4	Bazar	KING	Stephen	20	Américaine	Épouvante	Adolescent
4	Les Tommyknockers	KING	Stephen	20	Américaine	Épouvante	Adolescent
13	Sonnettes	KING	Stephen	20	Française	Fantastique	Adolescent
13	La mortte ensoumise	GAULTIER	Thérèse	19	Française	Épouvante	Adolescent
2	Le crime de l'orient express	CHRISTIE	Agatha	20	Anglaise	Policier	Adolescent
398	Les dix petits nègres	CHRISTIE	Agatha	20	Anglaise	Policier	Adolescent
8	Les mystères	HUGO	Victor	19	Française	Roman	Adolescent
22	Le Fleau	KING	Stephen	20	Américaine	Roman	Adolescent
4	Les Fontaines à l'annexe	HECHER	Charles	20	Française	Documentaire	Adulte
1	Les Fleurs du Mal	BAUDELAIRE	Charles	20	Française	Poésies	Adulte
15	La légende des siècles	HUGO	Victor	19	Française	Poésies	Adulte
28	Les hommes men amour	DURAS	Marguerite	20	Française	Roman	Adulte
18	L'annet	DURAS	Marguerite	20	Française	Roman	Adulte
6	Les bon petits diables	SEGOL	Constance	20	Française	Roman	Adulte
1	Mémoires d'Hadrien	YOURCENAR	Marguerite	20	Française	Roman	Adulte
19	Rajazer	RACINE	Jean	17	Française	Théâtre	Adulte
18	Les Incas	HERGE	Jean	20	Belge	Bande dessinée	Enfant
18	Obijesth l'unc	HERGE	Jean	20	Belge	Bande dessinée	Enfant
18	Trois au Congo	HERGE	Jean	20	Belge	Bande dessinée	Enfant

Dans le groupe **Police**, cliquez sur l'outil **Quadrillage** et choisissez le type de bordures à appliquer.

Menu :



Résultat obtenu :

Code Livre	Titre de l'ouvrage	Nom/Auteur	Prénom/Aut.	Epoque	Nationalité	Genre	Catégorie
30	20 000 lieues sous les mers	VERNE	Jules	19	Française	Aventure	Adolescent
30	Trois et les oranges bleues	VERNE	Jules	19	Belge	Bande dessinée	Adolescent
14	Caroline	VOITARE		18	Française	Classique	Adolescent
18	Zelig	VOITARE		18	Française	Classique	Adolescent
4	Bazar	KING	Stephen	20	Américaine	Épouvante	Adolescent
4	Les Tommyknockers	KING	Stephen	20	Américaine	Épouvante	Adolescent
13	Sonnettes	KING	Stephen	20	Française	Fantastique	Adolescent
13	La mortte ensoumise	GAULTIER	Thérèse	19	Française	Épouvante	Adolescent
2	Le crime de l'orient express	CHRISTIE	Agatha	20	Anglaise	Policier	Adolescent
398	Les dix petits nègres	CHRISTIE	Agatha	20	Anglaise	Policier	Adolescent
8	Les mystères	HUGO	Victor	19	Française	Roman	Adolescent
22	Le Fleau	KING	Stephen	20	Américaine	Roman	Adolescent
4	Les Fontaines à l'annexe	HECHER	Charles	20	Française	Documentaire	Adulte
1	Les Fleurs du Mal	BAUDELAIRE	Charles	20	Française	Poésies	Adulte
15	La légende des siècles	HUGO	Victor	19	Française	Poésies	Adulte
28	Les hommes men amour	DURAS	Marguerite	20	Française	Roman	Adulte
18	L'annet	DURAS	Marguerite	20	Française	Roman	Adulte
6	Les bon petits diables	SEGOL	Constance	20	Française	Roman	Adulte
1	Mémoires d'Hadrien	YOURCENAR	Marguerite	20	Française	Roman	Adulte
19	Rajazer	RACINE	Jean	17	Française	Théâtre	Adulte
18	Les Incas	HERGE	Jean	20	Belge	Bande dessinée	Enfant
18	Obijesth l'unc	HERGE	Jean	20	Belge	Bande dessinée	Enfant
18	Trois au Congo	HERGE	Jean	20	Belge	Bande dessinée	Enfant

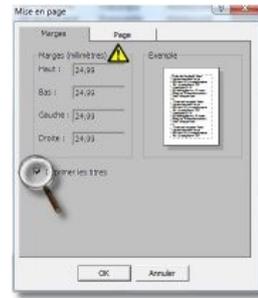
Dans le groupe **Police**, cliquez sur l'outil **Autres couleurs d'arrière-plan/remplissage** et choisissez dans la palette la couleur à appliquer. Cet outil va faciliter la mise en évidence des lignes en appliquant une couleur alternative sur deux lignes.

6.2. Mise en Page et Impression

Après avoir préparé la mise en forme, nous allons faire un peu de mise en page.

6.2.1. Versions Antérieures à 2007

Cliquez sur le menu **FICHIER** puis sur la commande **Mise en Page...**
Une boîte de dialogue à deux onglets s'affiche.



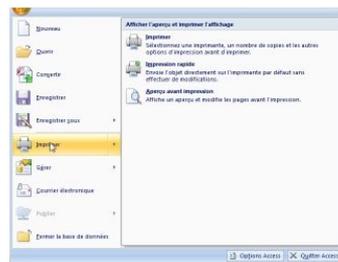
Tout d'abord l'onglet "**Marges**".
Pour cette boîte de dialogue, je ferai deux remarques :
1. les marges sont indiquées en **millimètres** ;
2. l'option indiquée par la loupe sert à imprimer le contenu sur un papier préimprimé.
Ceci ne nous préoccupe pas pour l'instant.



Puis l'onglet "**Pages**".
Comme précédemment deux remarques s'imposent pour cette boîte de dialogue :
1. par défaut l'orientation "**Portrait**" est activée.
2. il est possible de choisir une imprimante spécifique.

Nous ne pourrons pas alterner deux couleurs comme dans Access 2007.

6.2.2. Version Access 2007



Pour avoir accès à l'aperçu avant impression, il faut cliquer sur le bouton **Office**.
Dans la fenêtre affichée, cliquez
1. **Imprimer** ;
2. **Aperçu avant impression**.

Access affiche alors la fenêtre ci-dessous qui nous permettra en quelques clics de mettre en page notre document.



Cette image nous indique cinq repères.

1. Permet de changer l'orientation du document à l'impression (le logo nous indique la signification des expressions Portrait ou Paysage).
2. Permet de choisir la façon dont l'aperçu doit afficher les pages à imprimer à l'écran (donne une vue d'ensemble du résultat imprimé).
3. Permet de choisir des marges prédéfinies.
4. Affiche la boîte de dialogue de mise en page afin de personnaliser les marges (attention, les marges sont indiquées en **millimètres**).
5. Lance l'impression.

Dans la loupe, vous pouvez remarquer que le curseur s'est transformé en loupe contenant un signe +.

Il vous suffit de cliquer sur un point de l'aperçu pour provoquer un zoom avant à partir de ce point. Vous

remarquerez alors que le signe + se transforme en signe -. Il suffit alors de cliquer pour faire un zoom arrière.

7. CONCLUSION

Ce premier volet vous permet donc de créer vos requêtes, de les trier et d'imprimer le résultat sans vous lancer dans la génération d'un état. Dans le second volet nous aborderons l'utilisation des opérateurs dans les critères de sélection.

8. TÉLÉCHARGEMENT

Vous pouvez télécharger la base **Bouquins** afin de vous amuser à faire quelques requêtes sans risque : [Lien 129](#).

Retrouvez l'article de Jean Ballat en ligne : [Lien 130](#)



Bien débiter en C++

Vous souhaitez apprendre le C++ mais ne savez pas comment vous y prendre, par quoi commencer ? Cet article est fait pour vous ! Des outils les plus adaptés jusqu'aux meilleures ressources pour apprendre, cet article vous donnera toutes les pistes nécessaires pour effectuer un apprentissage correct du C++.

1. Qu'est-ce que le C++ ?

Le C++ est un langage de programmation très répandu et très utilisé : on s'en sert pour réaliser des programmes embarqués dans des appareils mobiles comme pour des jeux vidéo, des logiciels de gestion de données comme des compilateurs pour d'autres langages de programmation. Le C++ est un langage où l'on peut plus ou moins tout faire, et de différentes façons, ce qui fait sa puissance mais en le rendant de fait plus difficile à apprendre.

C'est au début des années 80 que Bjarne Stroustrup a inventé le "C with classes" : à ce moment-là, il ne s'agissait que d'un C auquel on avait rajouté des classes et de l'héritage. Depuis plus de 20 ans, le langage C++ a énormément évolué et il a été normalisé. Aujourd'hui, C et C++ sont deux langages très, très différents bien qu'ils partagent un peu d'histoire. C'est pourquoi dans cette page nous vous proposons des ressources qui vous permettront d'apprendre le C++ "moderne", et non plus en tant que langage qui hérite du C avec un apprentissage du C au préalable. En effet, il est même déconseillé pour un débutant d'apprendre le C avant le C++ si c'est pour apprendre le C++ au final.

Le C++ est un langage multiparadigmes : objet, fonctionnel, générique sont très communs en C++. Il est d'une richesse incroyable mais il faut connaître au mieux les règles du jeu pour écrire des logiciels solides, que l'on pourra faire évoluer facilement.

2. Les livres C++

Il n'existe pas **un** livre C++ qui convient à tous les profils, en revanche, nous vous en présentons ici selon différents critères, afin de choisir celui qui vous conviendra le mieux et qui rendra votre apprentissage le plus efficace possible. Vous trouverez une liste de livres C++ sur la page livres C++ : [Lien 131](#).

- Si vous n'avez jamais programmé avant, ou presque pas, nous vous conseillons *C++ je me lance* ([Lien 132](#)) de Francis Glassborow, bien qu'il date un peu il s'avère très efficace pour les débutants en programmation. Il se complètera ensuite très bien avec un ouvrage de référence sur le C++.
- *Programmation - Principes et pratique avec C++* ([Lien 133](#)) de Bjarne Stroustrup est un excellent livre pour apprendre à programmer que vous soyez grand débutant ou ayez un niveau intermédiaire.
- Si vous avez déjà programmé et que vous lisez l'anglais, vous pouvez vous tourner vers

Accelerated C++ ([Lien 134](#)) de Koenig & Moo pour une introduction rapide ou *C++ Primer, 4e édition* ([Lien 135](#)) pour une introduction plus souple.

3. Les cours en ligne sur le C++

Il y a un certain nombre de cours en ligne de bonne qualité : [Lien 136](#). Certains utilisent l'approche historique : on apprend le C++ en tant que sorte "d'évolution" du C, qui n'est plus l'approche appréciée aujourd'hui. De plus, un cours, aussi bon soit-il, ne remplacera pas un bon livre.

Deux des cours hébergés sur Developpez sont assez complets, malgré le fait qu'ils utilisent l'approche historique (ce qui vous demandera de garder un certain recul par rapport aux pratiques du C qu'ils peuvent reprendre) : le *mégacours* de Christian Casteyde ([Lien 137](#)) ainsi que *Penser en C++* (qui est la traduction d'un livre) de Bruce Eckel ([Lien 138](#)). Je recommanderais personnellement *Penser en C++* que je trouve légèrement plus correct et précis, mais ces deux cours vous formeront très convenablement aux bases du C++, vous permettant de vous reposer sur de bonnes bases pour la suite de votre apprentissage.

Pour terminer, quelles que soient les sources où vous puiserez vos informations, il vous faudra garder un certain recul et garder le meilleur de chacune car *La vérité n'est nulle part*.

4. Ce n'est pas suffisant ?

Non, un cours et/ou un livre sur le C++ ne sont pas suffisants. En effet, ces derniers répondent au problème du "comment" (comment écrire un logiciel qui fait ceci ou cela), alors qu'il y a aussi le problème du "pourquoi". Il vous faudra impérativement, si vous êtes un débutant total en programmation, vous munir d'un ou plusieurs ouvrages et/ou cours sur :

- l'algorithmique et les structures de données : il vous faudra souvent, lorsque vous programmerez, stocker des données de manière intelligente et efficace, mais surtout appropriée à l'usage qui en est fait. Connaître les structures de données les plus utilisées ainsi que leurs caractéristiques est un élément très important. De même, connaître les bases de l'algorithmique (variables, boucles, complexité, invariants...) est tout aussi nécessaire pour que vous écriviez de bons programmes. Vous pouvez vous référer à la page Algorithmique pour ce faire : [Lien 139](#) ;
- la conception logicielle : il vous faudra savoir

organiser intelligemment les différentes parties de vos programmes de sorte à pouvoir facilement introduire des évolutions dans ces derniers, par exemple. Pour ce faire, il y a également une certaine connaissance à acquérir et à mettre en pratique. Vous pouvez vous référer à la page Conception pour ce faire : [Lien 140](#).

5. Les outils C++

Le C++ existe depuis plus de 20 ans. Par conséquent, il y a une quantité phénoménale d'outils pour écrire du code C++, le compiler, le déboguer, le documenter, l'analyser, etc. Voici donc les recommandations de l'équipe à ce sujet. Ne sont présentés ici que les outils gratuits ; vous pouvez trouver une liste plus complète et incluant les produits payants sur la page outils : [Lien 141](#).

- Environnement de développement intégré (compilateur + éditeur de code source, au minimum, qui vous mâche le travail dont vous n'avez en général pas à vous soucier) : Microsoft Visual C++ Express (Windows seulement) ([Lien 142](#)) ou Code::Blocks (Windows, Linux, Mac) ([Lien 143](#)).
- Éditeur de texte (permet simplement d'écrire du code, souvent avec de la coloration syntaxique) : Notepad++ (Windows) ([Lien 144](#)), Éditeurs de texte préfourmis comme Emacs, VI(M), Kate, Gedit (Linux - selon la distribution).
- Compilateur (transforme du code C++ en code compréhensible par votre ordinateur) : Visual C++ (Windows) ([Lien 145](#)), MinGW (Windows) ([Lien 146](#)), Intel C++ Compiler (Windows, Linux) ([Lien 147](#)) ou g++ (Linux) ([Lien 148](#)).
- Débogueur (permet d'inspecter en détail l'exécution de vos programmes, trop peu utilisé par les débutants bien que très utile) : celui intégré à Visual C++ (Windows), GNU GDB (Linux) ([Lien 149](#)).
- Générateur de documentation (génère un ensemble de fichiers HTML depuis votre code ainsi que les commentaires qui l'accompagnent) : Doxygen ([Lien 150](#)).

Il vous est conseillé, dans un premier temps, d'utiliser un éditeur de code et le compilateur à côté. En effet, cela vous évitera de nombreux problèmes une fois que vous entamerez des manipulations plus compliquées que de connaître, au moins dans les grandes lignes, le fonctionnement de la compilation d'un programme C++, les options essentielles du compilateur, etc.

6. Il me reste des questions...

Alors vous avez bien appris grâce à un bon livre et/ou un bon cours, mais il vous reste des questions ? Dans ce cas, nous vous recommandons d'aller consulter la Foire Aux Questions C++ ([Lien 151](#)) qui aborde des sujets divers et variés à propos du C++, depuis les petites astuces de base jusqu'à des explications profondes sur les mécanismes du C++. Il y a également de nombreux tutoriels sur le C++ ([Lien 136](#)) qui abordent des aspects spécifiques afin de combler les lacunes de certains cours ou livres sur des points bien précis du langage. Enfin, si vous n'avez toujours pas trouvé de réponse à vos questions, n'hésitez

pas à vous rendre sur le forum C++ ([Lien 152](#)), lieu d'entraide convivial, pour nous faire part de vos difficultés et de vos questions.

Par ailleurs, c'est une mauvaise idée de se lancer dans le C++ en ayant comme objectif immédiat de programmer un clone du dernier jeu à la mode. Bien qu'il y ait de fortes chances qu'il soit effectivement écrit en C++, il a probablement demandé des années de travail à des équipes de programmeurs C++ expérimentés. Il faut apprendre petit à petit, mettre en application ce que l'on apprend, faire un effort de compréhension.

7. Les bibliothèques

Le C++ n'est pas comme le Visual Basic ou similaire : au début, tout ce que vous aurez, c'est une console. De base, en C++, on ne peut pas ouvrir une fenêtre ou envoyer un email. Il faut pour cela faire appel à ce que l'on appelle des bibliothèques. Ce sont des ensembles de fonctions et classes écrites en C++ qui vous permettront d'effectuer des tâches précises. Nous vous recommandons aussi de faire l'effort de vous concentrer sur le langage pendant un moment avant de vouloir créer des applications 3D ou autres clients de messagerie instantanée... Car généralement ces bibliothèques sont écrites par des personnes expérimentées et il n'est pas trivial de s'en servir correctement, malgré les documentations qui les accompagnent. Une fois que vous aurez un niveau satisfaisant en C++, vous pourrez consulter la page Bibliothèques C++ ([Lien 153](#)) puis la page Cours sur les bibliothèques C++ ([Lien 154](#)) pour vous former.

8. Je voudrais aller plus loin !

Vous aimez le langage et voudriez comprendre plus en profondeur comment tout cela fonctionne ? Vous souhaitez découvrir des techniques avancées, faire en sorte que le langage n'ait plus de secrets pour vous ? Alors voici quelques liens qui vous aideront à acquérir une compréhension profonde du C++.

FAQ

- La F.A.Q C++ aborde également un certain nombre de sujets avancés : [Lien 151](#).
- La FAQ Comeau Computing, en anglais : [Lien 155](#).
- C++ FAQ Lite, en anglais ([Lien 156](#)), et sa traduction partielle : [Lien 157](#).
- La FAQ de Bjarne Stroustrup, créateur du langage ([Lien 158](#)) et plus généralement sa page dédiée aux ressources sur le C++, en anglais : [Lien 159](#).

Articles

- Les tutoriels C++ abordent pour certains des techniques et sujets assez avancés sur le C++ : [Lien 137](#).
- Artima > C++ Source : contient de nombreux articles, en anglais, de grands noms du C++ sur des sujets généralement assez avancés : [Lien 160](#).
- Dr Dobb's Journal > C++ : même chose que le précédent : [Lien 161](#).

Voici de plus quelques livres qui décortiquent certains aspects du C++ de manière très intéressante et que nous vous recommandons.

Livres

- La série *Effective C++*, *More Effective C++* et *Effective STL* de Scott Meyers : elle aborde des aspects-clés du langage C++, tels que la bibliothèque standard, la programmation orientée objet, les exceptions, etc. Le but est de montrer les erreurs à ne pas faire et d'enseigner les bonnes pratiques du C++.
- La série *Exceptional C++*, *More Exceptional C++* and *Exceptional C++ Style* de Herb Sutter : sensiblement de même nature que la série d'ouvrages de Scott Meyers.
- *C++ Coding Standards*, de Herb Sutter et Andrei Alexandrescu : cet ouvrage enseigne les bonnes lignes de conduite en C++, sans trop rentrer dans les détails comme cela est fait dans les deux séries d'ouvrages ci-dessus.
- *Modern C++ Design* de Andrei Alexandrescu et *C++ Templates : The Complete Guide* de David Vandevor et Nicolai Josuttis abordent la conception générique en C++ et vous présentent en profondeur les templates du C++.
- *Multi-Paradigm Design for C++*, de James O. Coplien : montre comment la combinaison des

deux paradigmes principaux de C++, la programmation orientée objet et la programmation générique, s'avère puissante et efficace pour le développement C++ quotidien.

Pour terminer, vous plonger dans du code écrit par d'autres (expérimentés si possible) ne pourra vous être que bénéfique, afin d'élargir votre vision du C++, d'apprendre des petites techniques qui font la différence, etc. Regarder le code de bibliothèques C++ ou bien de petits logiciels ou jeux vous sera ainsi très bénéfique.

9. Conclusion

Vous avez désormais toutes les cartes en main pour effectuer un apprentissage du C++ sur mesure, efficace. Comme pour l'apprentissage de n'importe quel langage, nous vous conseillons de mettre en pratique autant que nécessaire pour votre compréhension. Enfin, n'oubliez pas que le C++ est riche et par conséquent complexe, donc peut-être ressentirez-vous des difficultés à certains moments ; toutefois, cela fait sa force et c'est pourquoi il est important de surmonter ces difficultés.

Retrouvez l'article d'Alp Mestan en ligne : [Lien 162](#)

Microsoft Visual C++ 2010 Express : Téléchargement, installation et configuration

Ce tutoriel présente la procédure d'installation et de configuration de Visual C++ Express Edition 2010, l'environnement de développement C++ gratuit de Microsoft. Poursuivant la même politique de mise à disposition d'une version gratuite, la sortie de Visual Studio 2010 s'est accompagnée des versions Express : Visual C++ 2010, Visual Basic 2010, Visual C# 2010, Visual Web Developer 2010 et Visual Studio 2010 Express for Windows Phone CTP. Nous abordons ici l'installation de Visual C++ Express 2010. Visual C++ Express 2010 constitue un environnement de développement complet en C et C++.

1. Avant-propos

Apparues en 2005, les versions Express de Visual Studio étendent la gamme des environnements de développement proposés par Microsoft. Gratuites et disponibles pour différents langages, elles offrent un environnement de développement suffisamment mature pour des projets personnels ou semi-professionnels.

Ainsi, la sortie de Visual Studio 2010 s'est tout naturellement accompagnée de celle de Visual C++ Express 2010. Bien sûr, le studio est moins complet que les versions payantes, mais le compilateur est réputé être le même et, de toute façon, reste un bon outil de développement sur PC Windows.

Vous trouverez un tableau des différentes versions sur le site de Microsoft : [Lien 163](#).

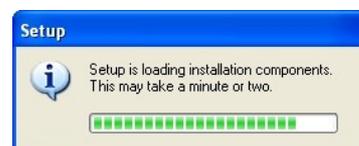
2. Installer Visual C++ Express 2010

2.1. Téléchargement

Visual C++ Express est disponible au téléchargement sur le site de Microsoft : [Lien 164](#). Pour réaliser cette présentation, nous avons téléchargé la version anglaise de Visual C++ Express 2010. Notre point de départ est par conséquent : vc_web.exe !

2.2. Installation

L'installation peut démarrer en exécutant le fichier que nous venons de télécharger : double-cliquez sur vc_web.exe. Ce petit fichier de 3 Mo n'est pas encore l'installateur à proprement parler. Il déclenche le téléchargement des composants nécessaires à l'installation du produit :



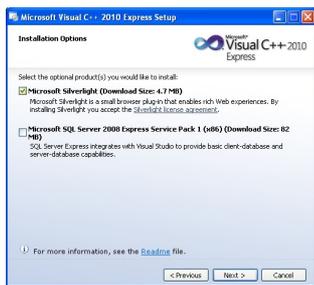
Une fois ces premiers téléchargements terminés, un assistant va nous guider tout le long de l'installation :



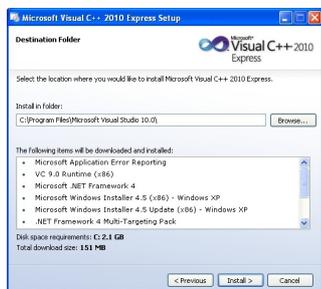
Après avoir cliqué sur suivant, nous arrivons sur les termes du contrat de la licence :



Veillez à bien lire les différents éléments et à en comprendre la teneur. Ceci fait, indiquez-le à l'assistant et passez à l'étape suivante :



Lors de l'installation de Visual C++ Express 2010, il vous est proposé d'installer également Microsoft Silverlight et Microsoft SQL Server 2008 Express Service Pack 1. Nous avons choisi de ne pas installer SQL Server mais d'installer Silverlight. Vous devez donc cocher la case correspondant à Silverlight et décocher celle de SQL Server. Il faut choisir maintenant le répertoire de destination :



Outre le répertoire d'installation, cette boîte de dialogue nous donne deux informations intéressantes. D'abord, le volume de données téléchargées : ici un peu plus de 150 Mo. Nous allons donc devoir faire preuve de plus ou moins de patience selon la qualité de la connexion Internet...

Ensuite, la liste détaille explicitement tous les composants nécessaires au fonctionnement de Visual C++ Express 2010. Cette liste dépend des éléments éventuellement déjà présents. Pour votre installation, cette liste peut varier puisqu'elle est directement liée à la configuration de votre poste. Ici, nous sommes parti d'un PC neuf sur lequel tout ou presque était à télécharger et à installer. Par conséquent, ce sont pas moins de dix composants qui sont traités :

- Microsoft Application Error Reporting ;
- VC 9.0 Runtime (x86) ;
- Microsoft .Net Framework 4 ;
- Microsoft Windows Installer 4.5 (x86) - Windows XP ;

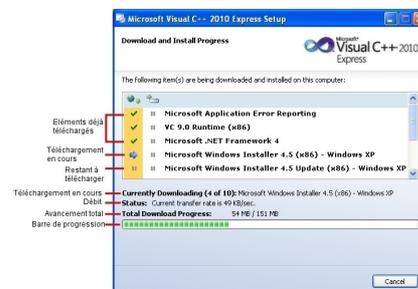
- Microsoft Windows Installer 4.5. Update (x86) - Windows XP ;
- .NET Framework 4 Multi-Targeting Pack ;
- Microsoft Visual C++ 2010 Express ;
- Microsoft SQL Server Compact 3.5 SP2 (x86) ENU ;
- Microsoft SilverLight ;
- Microsoft Help Viewer 1.0 x86.

Si d'aventure vous souhaitez désinstaller la version Visual C++ Express Edition 2010, il faut veiller à désinstaller ces différents produits à partir du gestionnaire d'ajout et de suppression des programmes. Mais terminons d'abord cette installation.

Une fois l'étape précédente franchie, la procédure se décompose en deux phases. Au départ, l'ensemble des composants est téléchargé :



Selon le volume total de données à télécharger et le débit de votre liaison Internet, ce téléchargement peut être plus ou moins long. La petite barre située à gauche de la liste indique les éléments déjà téléchargés (petite coche verte), l'élément en cours de téléchargement (la flèche bleue) et les éléments restant à télécharger (les deux points gris) :



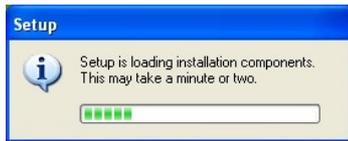
Une fois les téléchargements des différents composants terminés, l'installation commence. La même fenêtre d'avancement devrait éviter le dépaysement :



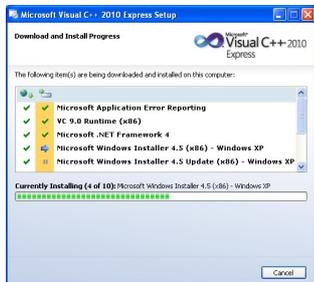
Selon votre environnement, certains composants peuvent nécessiter le redémarrage du système pour valider leur installation :



Cliquez simplement sur la demande de redémarrage. L'installateur arrête et redémarre le système. À cette reprise, l'installation repart par ce message :



Puis l'installation se poursuit avec la fenêtre d'avancement maintenant bien connue :



Une fois la liste des composants égrenée, l'installation de Visual C++ Express 2010 se termine :



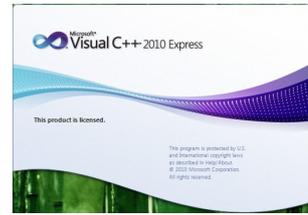
3. Premier lancement et enregistrement

Il est temps de lancer Visual C++ Express 2010. L'icône peut se trouver sur votre bureau, sur la barre de lancement rapide ou dans le menu démarrer. Si vous ne la trouvez pas, le programme de lancement se trouve (en fonction du répertoire d'installation) en suivant ce chemin :

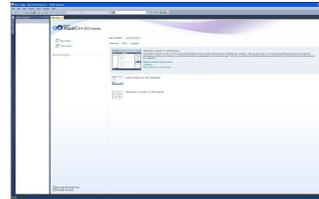
- C:\Program Files\Microsoft Visual Studio 10.0\Common7\IDE.

L'exécutable lançant le studio de développement est VCEXpress.exe.

Le démarrage de Visual C++ Express 2010 est légèrement plus lent que son prédécesseur Visual C++ Express 2008 mais cela reste très raisonnable. Cela commence par l'affichage de l'image suivante :

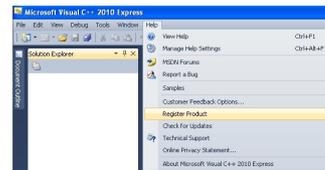


Puis voilà la première page affichée :

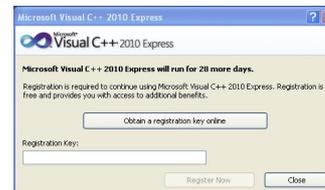


À ce moment de l'installation, Visual C++ Express 2010 n'est valide que pour 30 jours. Pour pérenniser cette installation et pouvoir bénéficier de cet outil sans durée de limitation, vous devez vous enregistrer auprès de Microsoft. Cet enregistrement est gratuit et n'aboutit pas à remplir votre boîte aux lettres de publicité.

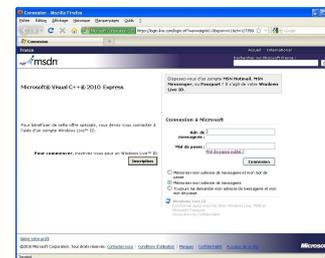
Allez dans le menu Help (Aide) et choisissez l'option d'enregistrement du produit :



Une boîte de dialogue vous invite alors à saisir une clé d'enregistrement. Pour l'obtenir, il faut appuyer sur le bouton dédié à cet effet :



Votre navigateur internet s'ouvre alors sur la page d'enregistrement :



Vous vous connectez avec votre identifiant MSN. Si vous n'en avez pas, vous pouvez en créer un.

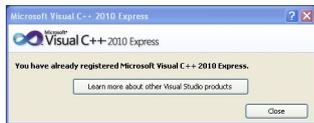
Une fois connecté, le site vous fournit votre identifiant. Sélectionnez-le et copiez-le dans le presse-papier :



Il ne vous reste plus qu'à le coller dans la zone d'édition demandant la clé d'enregistrement et à valider :



Un message vient vous confirmer de la réussite de l'enregistrement :



4. Éléments de configuration

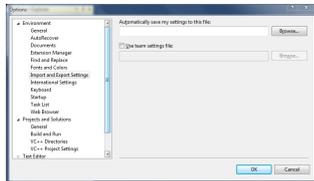
4.1. Configuration basique/expert

Première surprise : au premier lancement, le menu de Visual C++ Express 2010 semble bien allégé par rapport aux versions précédentes. L'I.D.E. propose de fonctionner en mode basic ou en mode expert. Pour faciliter la configuration, positionnons-nous en mode expert :



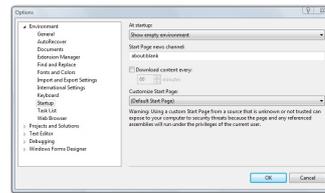
4.2. Sauvegarde des paramètres de configuration

Il est possible de spécifier le répertoire et le fichier où sont sauvegardés les paramètres de configuration :



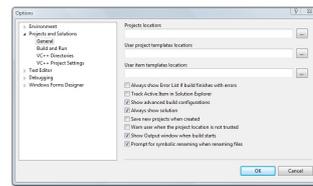
4.3. Page d'accueil

La page affichée au démarrage de Visual C++ Express 2010 peut être modifiée :



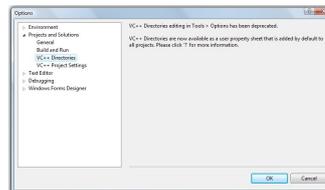
4.4. Répertoire des projets

Dans l'onglet *Général* des options de configuration pour les projets et solutions, vous pouvez spécifier le répertoire où Visual C++ Express 2010 doit créer ceux-ci :



4.5. Configuration des répertoires des bibliothèques

La configuration des répertoires de recherche des fichiers d'en-têtes (.h) pour les bibliothèques tierces ne peut plus se faire au niveau global de Visual C++ Express 2010 mais se situe désormais au niveau projet :



Voyons donc comment créer un nouveau projet.

Retrouvez la suite de l'article de 3DArchi en ligne : [Lien 165](#)

Débuter dans la création d'interfaces graphiques avec Qt 4

Ce tutoriel fournit une introduction à la programmation d'interfaces graphiques (IHM, interface homme-machine) utilisant la boîte à outils Qt. Il ne couvre pas tout ; l'accent est mis sur la philosophie de la programmation IHM, et les fonctionnalités de Qt sont introduites au fur et à mesure. Certaines fonctionnalités, bien qu'elles soient souvent utilisées, ne sont pas couvertes par ce tutoriel d'introduction à Qt.

1. Présentation

Ce tutoriel fournit une introduction à la programmation d'interfaces graphiques (IHM, Interface Homme-Machine) utilisant la boîte à outils Qt. Il ne couvre pas tout ; l'accent est mis sur la philosophie de la programmation IHM, et les fonctionnalités de Qt sont introduites au fur et à mesure. Certaines fonctionnalités, bien que souvent utilisées, ne sont pas couvertes par ce tutoriel d'introduction à Qt.

Le premier chapitre commence avec un programme minimal de type « hello world » et les suivants introduisent de nouveaux concepts, jusqu'au chapitre 14, où le « hello world » du premier chapitre va se transformer en un jeu de 650 lignes de code.

Si vous débutez complètement avec Qt, commencez par lire Comment apprendre Qt si vous ne l'avez pas déjà fait : [Lien 166](#).

Le code source du tutoriel est situé dans le répertoire exemples/tutorials/tutorial de Qt 4.4. Pour les autres versions, vous pouvez télécharger les sources sous la forme d'une archive RAR ([Lien 167](#)) ou TAR.BZ2 ([Lien 168](#)).

Chapitres :

1. *hello world* ;
2. terminer proprement l'application ;
3. valeurs usuelles ;
4. soyons widgets ;
5. les widgets ;
6. construire des widgets en vrac !
7. une chose mène à une autre ;
8. préparer le champ de bataille ;
9. avec un canon, on peut tout faire ;
10. doux comme un agneau ;
11. tire-lui dessus !
12. accrochons des briques en l'air ;
13. *game over* ;
14. face au mur.

Ce petit jeu ne ressemble pas beaucoup aux applications IHM modernes. Il utilise quelques techniques IHM, mais nous vous recommandons, quand vous aurez bien travaillé avec lui, de jeter un œil à l'exemple de l'Application ([Lien 169](#)), qui présente une petite application IHM, avec des menus, des barres d'outils, une barre de statut, et bien d'autres encore.

1.1. Installer Qt

La meilleure manière d'installer Qt est d'utiliser le Qt SDK. Vous pouvez le télécharger pour votre plateforme sur le site de Nokia : [Lien 170](#). Ainsi, vous aurez une installation complète de Qt, ce qui comprend toutes les bibliothèques qui constituent Qt, un compilateur (GCC/MinGW) et un environnement de compilation (Qt Creator).

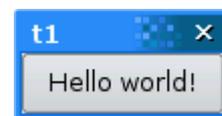
2. Hello World

Sources : [Lien 171](#).

Fichiers

- tutorials/tutorial/t1/main.cpp
- tutorials/tutorial/t1/t1.pro

Ce premier programme est un simple exemple de Hello World. Il contient juste le minimum pour créer une application Qt. Voici un aperçu du programme :



Et voilà le code source complet de l'application :

```
#include <QApplication>
#include <QPushButton>

int main(int argc, char *argv[])
{
    QApplication app(argc, argv);

    QPushButton hello("Hello world!");

    hello.show();
    return app.exec();
}
```

2.1. Analyse du code ligne par ligne

```
#include <QApplication>
```

Cette ligne inclut la définition de la classe QApplication. Il doit y avoir une instance de QApplication dans toute application qui possède une IHM avec Qt. QApplication gère différentes ressources au niveau de l'application, comme la police de caractères par défaut ou le curseur.

```
#include <QPushButton>
```

Cette ligne inclut la définition de la classe QPushButton. Pour chaque classe faisant partie de l'API publique de Qt, il y a un fichier d'en-tête de même nom contenant sa définition. QPushButton est un bouton d'interface graphique que l'utilisateur peut presser et relâcher. Il gère lui-même son propre *look and feel*, comme n'importe quel QWidget. Un widget est un composant de l'interface graphique qui peut gérer une entrée de l'utilisateur et dessiner. Le programmeur peut changer tout le *look and feel* du widget ou seulement une partie de celui-ci (la couleur, par exemple), ainsi que son contenu. Un QPushButton peut afficher du texte ou un QIcon.

```
int main(int argc, char *argv[]){
```

La fonction main() est le point d'entrée du programme. Le plus souvent, l'utilisation de cette fonction ne nécessite que quelques initialisations avant de passer la main à la bibliothèque Qt, qui indique ensuite au programme, par le biais d'événements, les actions demandées par l'utilisateur. Le paramètre argc est le nombre d'arguments de la ligne de commande et argv est le tableau contenant les arguments de la ligne de commande. C'est une propriété standard du C++.

```
QApplication app(argc, argv);
```

app est l'instance de QApplication de ce programme. Elle est créée ici. Les arguments argc et argv sont passés au constructeur de QApplication pour qu'il puisse traiter certains arguments de ligne de commande standard (comme -display sous X11). Tous les arguments de ligne de commande reconnus par Qt sont enlevés de argv, et argc est décrémenté en conséquence. Voir la documentation de QApplication::arguments() pour plus de détails : [Lien 172](#). L'instance de QApplication doit être créée avant toute utilisation d'une quelconque fonctionnalité d'IHM de Qt.

```
QPushButton hello("Hello world!");
```

Après la création de la QApplication, voilà la première ligne de code liée à l'IHM : un bouton est créé. Le bouton affiche le texte "Hello world!". Comme, dans le constructeur de QPushButton, on n'a pas spécifié un second argument pour préciser la fenêtre parente du bouton, ce bouton sera lui-même une fenêtre, avec son propre cadre et sa barre de titre. La taille du bouton est déterminée par défaut. On pourrait appeler QWidget::move() pour assigner un emplacement spécifique au widget, mais nous laissons ici le système choisir une position.

```
hello.show();
```

Un widget n'est jamais visible quand on le crée. Il faut appeler QWidget::show() pour le rendre visible.

```
return app.exec();}
```

C'est ici que le main() passe le contrôle à Qt. QCoreApplication::exec() finira en même temps que

l'application. QCoreApplication est une classe de base de QApplication. Elle implémente les fonctionnalités non liées à l'IHM et peut être utilisée pour le développement d'applications sans IHM. Dans QCoreApplication::exec(), Qt reçoit et traite les événements système et les renvoie aux widgets appropriés. Vous devriez maintenant compiler et lancer le programme.

2.2. Les variables d'environnement

Si vous utilisez la ligne de commande pour développer des applications Qt, vous devez vous assurer que les bibliothèques et les exécutables de Qt sont accessibles à votre environnement en ajoutant le chemin du répertoire bin de Qt à votre variable PATH. Cette opération est décrite dans les instructions d'installation de votre plate-forme. Sous Windows, ceci est automatique si vous utilisez la fenêtre de commande depuis le menu Démarrer > (Tous les) Programmes > Qt. Si vous utilisez la fenêtre de commande depuis Démarrer > Exécuter > command ou cmd, vous devrez paramétrer vous-même la variable PATH ou utiliser le script du Qt SDK qtenv.bat, comme décrit ci-après.

2.3. Compiler l'application

Les exemples de ce tutoriel sont situés dans le répertoire examples/tutorials/tutorial de Qt. Si vous avez installé Qt à partir d'un exécutable, les exemples ont également été installés. Si vous avez compilé Qt vous-même, les exemples ont été compilés en même temps. Dans les deux cas, en modifiant et en recompilant vous-même ces exemples, vous en apprendrez beaucoup sur la façon d'utiliser Qt. Dans le cas où vous auriez copié le(s) fichier(s) .cpp et .h d'un exemple dans un autre répertoire pour y apporter vos modifications, la prochaine étape sera de créer un makefile pour Qt dans ce répertoire. La création d'un makefile pour Qt utilise la commande qmake, un outil de compilation fourni avec Qt. Pour créer un makefile, lancez les deux commandes suivantes dans le répertoire contenant vos sources modifiées :

```
qmake -project
qmake
make # ou nmake
```

La première commande ordonne à qmake de créer un fichier de projet (.pro). La seconde commande ordonne à qmake d'utiliser le fichier .pro pour créer un makefile adapté à la plateforme et au compilateur. Vous n'avez plus qu'à lancer make (nmake si vous êtes sous Visual Studio) pour compiler le programme et vous pourrez ensuite lancer votre première application Qt !

Il n'y a pas d'emplacement spécifique pour vos fichiers, c'est à vous de le déterminer en fonction de votre système, qmake s'y adaptera. Il suffit de le lancer à la racine de vos sources.

Il est possible que vous deviez spécifier une variable d'environnement, QMAKESPEC, qui correspond à votre plateforme (le triplet système d'exploitation, compilateur, bitness).

De même, il est probable que vous deviez modifier votre PATH avant d'appeler qmake, sous peine d'erreurs, sous Windows. Si vous utilisez le Qt SDK, vous devriez utiliser

le script qtenv.bat contenu dans le dossier Dossier d'installation du Qt SDK\bin, soit généralement un chemin du genre C:\Qt\2010.05\bin. Ce script définira aussi la variable QMAKESPEC en fonction de ce qui est nécessaire pour votre environnement.

2.4. Lancer l'application

Quand vous lancerez cet exemple, vous verrez une petite fenêtre affichant un seul bouton. Sur le bouton, vous pourrez lire le fameux : "Hello world!".

Vous pourrez trouver l'exécutable *release* dans le dossier *release* à la racine de vos sources.

2.5. Exercices

Essayez de dimensionner la fenêtre. Si vous êtes sous X11, essayez de lancer le programme avec l'option `-geometry` (par exemple, `-geometry 100x200+10+20`).

Retrouvez la suite de l'article de nokia traduit par l'équipe Qt de Developpez.com en ligne : [Lien 173](#)

PySide Simplissimus

PySide Simplissimus : des tutoriels d'une dizaine de minutes pour un débutant par un débutant.

1. Les articles originaux

Le Qt Developer Network est un réseau de développeurs utilisant Qt afin de partager leur savoir sur ce framework. Vous pouvez le consulter en anglais : [Lien 174](#).

Nokia, Qt, Qt Quarterly et leurs logos sont des marques déposées de *Nokia Corporation* en Finlande et/ou dans les autres pays. Les autres marques déposées sont détenues par leurs propriétaires respectifs.

Cet article est la traduction de PySideSimplissimus : 10 min newbie to newbie Tutorials ([Lien 175](#)), PySideSimplissimus Module 1 Prerequisites ([Lien 176](#)), Setting up PySide ([Lien 177](#)), Your first PySide application ([Lien 178](#)).

2. Table des matières

2.1. Prérequis

Avant de commencer à programmer, il y a des prérequis. Dans cette section dédiée, quelques liens pour aller plus loin ainsi qu'un exemple *Hello World*.

À la suite sont proposés quelques petits exemples. À moins que ça ne soit précisé, ils devraient prendre une dizaine de minutes. Cela pourrait prendre plus de temps, en fonction de divers facteurs, dont la connaissance de PySide.

2.2. Suite des tutoriels

Fermer : [Lien 179](#)

À propos de : [Lien 180](#)

Montrer la licence : [Lien 181](#)

Combiner : [Lien 182](#)

Combinaisons alternatives : [Lien 183](#)

Combiner tout en un : [Lien 184](#)

Une application d'ingénierie (presque) pratique : [Lien 185](#)

3. Prérequis

L'exemple le plus important quand on apprend un langage est l'éculé « Hello World ». Il peut aussi occasionnellement être le plus difficile de tous, s'il n'y a pas de tutoriel déjà prêt. Même pour un Hello World, il y a des prérequis ! Tout d'abord, on doit installer PySide.

3.1. Installer PySide

Avant de jouer avec PySide, il faut installer ce qui suit :

- Qt : [Lien 186](#) ;
- Python : [Lien 187](#) ;
- PySide : [Lien 188](#).

Pour vérifier si PySide est bien installé, entrez simplement cette commande dans une console Python :

```
import PySide
```

Si aucune erreur n'est retournée, vous devriez avoir installé PySide avec succès et pouvez commencer à développer avec Qt et Python. Il est possible aussi de vérifier la version de PySide :

```
import PySide
print PySide.__version__
```

Le résultat devrait être quelque chose comme 1.0.0.

3.2. Hello World

Une version de PySide étant installée sur votre ordinateur, vous pouvez commencer à développer des applications avec Qt en Python. Comme tout autre framework de programmation, commençons par le traditionnel Hello World.

Voici un exemple simple de Hello World dans PySide :

```
#!/usr/bin/pythonimport sys
from PySide.QtCore import *
from PySide.QtGui import *

app = QApplication(sys.argv)
# Create a Label and show it
label = QLabel("Hello World")
label.show()
```

```
# Enter Qt application main loop
app.exec_()
sys.exit()
```

Avec les applications desktop PySide, le fichier doit débuter en important PySide.QtCore et PySide.QtGui. Ces classes contiennent les fonctions principales pour construire des applications PySide. Par exemple, PySide.QtGui contient des fonctions pour gérer des widgets, alors que PySide.QtCore contient des méthodes pour gérer des signaux et des slots et contrôler l'application.

Après ces imports, on crée une QApplication, l'application Qt principale. Comme Qt peut recevoir des arguments depuis la ligne de commande, on peut passer n'importe quel argument à l'objet QApplication. Habituellement, on

n'a pas besoin de lui passer des arguments, on peut le laisser tel quel.

Après la création de l'objet de l'application, on crée un objet QLabel. Il s'agit d'un widget qui peut présenter du texte (simple ou riche, comme du HTML) et des images. Notez que, après la création du label, on peut appeler la méthode show() qui va présenter le label à l'utilisateur.

Finalement, on appelle app.exec_(), qui va entrer dans la boucle principale Qt et commencer à exécuter le code Qt. En réalité, c'est seulement à partir de ce moment que le label est montré, mais ce point peut être ignoré pour le moment.

Retrouvez l'article de nokia traduit par Thibaut Cuvelier en ligne : [Lien 189](#)

PySide Simplissimus : fermer

Ceci ne vous prendra pas plus de cinq minutes : le but de ce programme est de faire en sorte qu'un clic sur le pushButton arrête l'exécution de ce programme. Comme toujours, on commence dans le Qt Designer. On appelle le programme d'une manière plutôt fleurie, quitter, ainsi le designer crée une MainWindow avec un bouton qui est enregistré en tant que quitter.ui.

1. L'article original

Le Qt Developer Network est un réseau de développeurs utilisant Qt afin de partager leur savoir sur ce framework. Vous pouvez le consulter en anglais : [Lien 174](#).

Nokia, Qt, Qt Quarterly et leurs logos sont des marques déposées de *Nokia Corporation* en Finlande et/ou dans les autres pays. Les autres marques déposées sont détenues par leurs propriétaires respectifs.

Cet article est la traduction de Close : [Lien 190](#).

2. Fermer

Le fichier Qt Designer est disponible : [Lien 191](#). On peut le transformer en fichier Python exécutable :

```
pyside-uic quitter.ui -o ui_quitter.py
```

Une fois le code généré, on peut l'utiliser :

```
#!/usr/bin/env python
# quitter.py - provide a button to quit this
"program"

import sys

from PySide.QtGui import QMainWindow,
QPushButton, QApplication

from ui_quitter import Ui_MainWindow

class MainWindow(QMainWindow, Ui_MainWindow):
    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        self.setupUi(self)

if __name__ == '__main__':
    app = QApplication(sys.argv)
```

```
frame = MainWindow()
frame.show()
app.exec_()
```

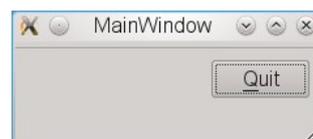
Juste pour que ce soit difficile, on a utilisé une procédure alternative pour créer une forme lisible par Python du fichier ui_quitter.py. Le nom de fichier est pratique, car il suffit de se souvenir d'un seul nom : quitter.

Note : si vous n'avez pas les setuptools installés, vous recevrez cette erreur lors de l'exécution de pyside-uic :

```
Traceback (most recent call last):
File "c:\Python26\Scripts\pyside-uic-script.py",
line 5, in <module>
    from pkg_resources import load_entry_point
ImportError: No module named pkg_resources
```

Les setuptools sont librement disponibles : [Lien 192](#).

En lançant le programme, vous trouverez qu'il peut être fermé de la manière habituelle par la croix ou bien en cliquant sur le bouton. Voici une image de la GUI :



Maintenant, regardons de plus près le code. Où est le code qui fait que le bouton ferme le programme ? Ce n'est sûrement pas dans le listing.

La raison de cette absence est qu'il a été créé par Qt Designer, qui possède un éditeur de signaux et de slots. On l'a utilisé de manière furtive pour connecter le « signal » QPushButton.click() au « slot » MainWindow.close(). Il est inclus dans le fichier quitter.ui et, suite à la conversion,

dans `ui_quitter.py`. Depuis ce fichier, on importe la classe `Ui_MainWindow`, l'un des ancêtres de notre classe `MainWindow`. On utilise l'héritage multiple de Python pour faciliter l'utilisation des méthodes dans le module `ui_quitter.py`.

Il y a toujours d'autres manières de procéder et on pourrait éviter le double héritage, mais pourquoi ne pas l'utiliser quand il est bien commode ? L'hérité creuse de plus en plus profondément dans l'héritage et initialise l'ancêtre

immédiatement au-dessus de `MainWindow` et ainsi l'arbre tout entier.

Vous remarquerez que la méthode `init` de `MainWindow` spécifie l'ancêtre à `None`. C'est le genre d'indication que l'on donne à `PySide` pour dire que `MainWindow` est bien la fenêtre principale du programme.

Retrouvez l'article de nokia traduit par Thibaut Cuvelier en ligne : [Lien 179](#)

Les bases de PyQt

Cet article est réservé aux débutants. On va y découvrir ce qu'est PyQt, ce qu'il fait et les composants principaux qu'il vaut mieux connaître avant de l'utiliser.

1. Avertissement

Si vous êtes intéressé par PyQt mais sans savoir ce qu'il en est, il est préférable de lire l'introduction : [Lien 193](#).

2. Qu'est-ce que PyQt ?

PyQt est un module Python qui permet d'accéder à la bibliothèque Qt de Nokia.

Il s'agit d'un framework applicatif multiplateforme : une même application PyQt fonctionne sans problème sous OS X, Linux et Windows.

C'est aussi un gros module de plus de trois cents classes, de six mille fonctions.

Il fournit un ensemble complet de classes utiles qui fournissent des interfaces pour l'accès aux fichiers, parser du XML, des structures de données et plus, ainsi que des outils comme Qt Designer, Qt Assistant et Qt Linguist pour concevoir des interfaces, accéder à l'aider et faciliter les traductions.

C'est donc un framework complet : il fournit des classes qui gèrent les types basiques de données comme celles qui effectuent des tests unitaires.

PyQt est développé par Phil Thompson, chez Riverbank Computing Limited : [Lien 194](#).

3. Que comprend PyQt ?

Les composants majeurs de PyQt sont :

- le module `QtGui`, pour tout ce qui concerne les GUI ;
- le module `QtCore`, pour tout ce qui ne concerne pas les GUI ;
- l'espace de noms `Qt`, pour les types et autres identifiants ;
- les outils `Qt`, pour d'autres tâches variées.

Le paquet PyQt dans Python, une fois installé, peut être facilement importé :

```
import PyQt4
```

D'autres modules plus spécialisés sont aussi disponibles : `QtHelp`, `QtOpenGL`, `QtScript`, `QtXml`, `QtSql`, `QtTest` et

bien d'autres.

Des scripts pour les ressources, interfaces et traductions sont aussi fournis, pour compatibilité avec son pendant C++.

4. Le module QtCore

On peut l'importer directement :

```
from PyQt4 import QtCore
```

Il contient toutes les classes essentielles et multiplateformes qui forment le squelette de toute application PyQt. Elles vont des chaînes de caractères à la gestion des processus, des entrées et des sorties, ainsi que diverses structures de données.

Quelques exemples de classes :

- `QString` : elle stocke et manipule des chaînes de caractères ;
- `QFile` : elle effectue des opérations d'entrée et de sortie sur des fichiers ;
- `QProcess` : elle lance des programmes externes ;
- `QRect` : une classe géométrique de rectangle ;
- `QDate` : elle fournit des fonctions sur les dates.

On peut évidemment trouver la liste complète des classes de `QtCore` : [Lien 195](#).

5. Le module QtGui

On peut l'importer directement :

```
from PyQt4 import QtGui
```

Il contient tous les éléments GUI fournis par Qt, d'un simple label à la complexe vue graphique. Tous les éléments de GUI dans PyQt sont appelés des « widgets ».

Quelques exemples :

- `QLabel` : un simple widget affichant une image ou un texte riche ;
- `QLineEdit` : une zone d'entrée sur une ligne ;
- `QPushButton` : un bouton standard qui peut être pressé ;
- `QCheckBox` : une simple case à cocher, avec deux ou trois états.

QtGui étend de manière naturelle QtCore, en donnant à chacune de ses fonctionnalités une interface graphique.

On peut évidemment trouver la liste complète des classes de QtGui : [Lien 196](#).

6. L'espace de noms Qt

On peut l'importer directement :

```
from PyQt4.QtCore import Qt
```

Sinon, il est accessible via l'espace de noms QtCore.Qt.

Il contient tous les identifiants, aux objectifs divers, des comportements GUI pour la plupart.

Ces constantes fournissent de multiples options utilisables pour définir le comportement d'un widget, sa position, etc.

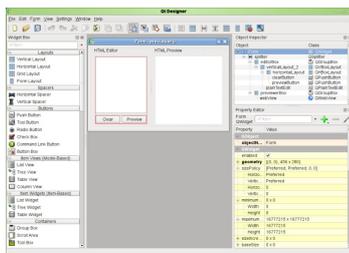
Quelques exemples :

- **Alignment** : pour aligner le texte sur un widget de différentes manières ;
- **Axis** : pour choisir l'axe X, Y ou Z pour une transformation ;
- **BrushStyle** : pour choisir un style de pinceau approprié ;
- **UIEffect** : pour définir des effets comme une animation, un dégradé, etc.

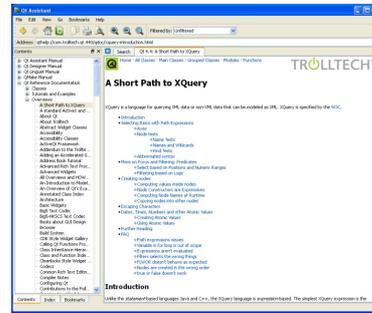
On peut évidemment trouver la liste complète des identifiants de Qt : [Lien 197](#).

7. Les outils de Qt

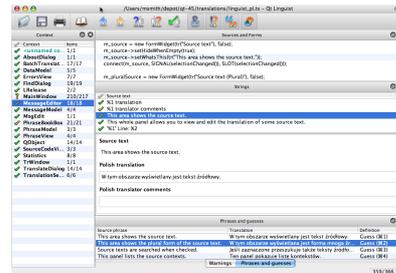
Qt Designer est prévu pour concevoir des GUI et les sauvegarder dans un format XML (.ui), qui peut alors être converti en fichier source Python avec le script pyuic4.



Qt Assistant fournit toutes les références de toutes les classes avec recherche, ainsi que d'autres documents utiles sur la bibliothèque Qt. Un outil très utile pour un programmeur.



Qt Linguist aide à traduire les chaînes des applications en différentes langues, fournissant ainsi une internationalisation facile d'un produit.



Tous ces outils sont installés avec le framework Qt, pas PyQt, qui ne fait que fournir des scripts pour les porter à Python.

8. Notes

La meilleure manière d'apprendre à utiliser Qt est de suivre la documentation, par exemple avec Qt Assistant, qui vient avec lui. Bien que prévue pour le C++, l'utilisation que l'on en fait en Python n'est pas très différente. Qt n'est pas seulement partout sur toutes les plateformes, il est aussi sur tous les langages, Ruby, Python, C++, Java et d'autres.

Si vous voulez vous référer à un livre, Rapid GUI Programming with Python and Qt, de Mark Summerfield ([Lien 198](#)) ou GUI Programming with Python: Qt Edition, de Boudewijn Rempt ([Lien 199](#)). Mon préféré, cependant, est C++ GUI Programming with Qt, de Jasmin Blanchette et Mark Summerfield ([Lien 200](#)). Bien qu'il soit en C++, il a une bonne approche.

Retrouvez l'article de Harsh traduit par Thibaut Cuvelier en ligne : [Lien 201](#)

Web sémantique

Tutoriels et articles



Introduction au Web sémantique

Nous allons voir ici ce qu'est le Web sémantique. Il faut savoir que ce domaine est tout neuf et encore trop peu connu. Je vais donc ici vous faire découvrir ce monde mystérieux.

1. Introduction

Le Web sémantique (plus techniquement appelé « le Web de données ») permet aux machines de comprendre la sémantique, la signification de l'information sur le Web. Il étend le réseau des hyperliens entre des pages Web classiques par un réseau de lien entre données structurées permettant ainsi aux agents automatisés d'accéder plus intelligemment aux différentes sources de données contenues sur le Web et, de cette manière, d'effectuer des tâches (recherche, apprentissage, etc.) plus précises pour les utilisateurs. Le terme a été inventé par Tim Berners-Lee, co-inventeur du Web et directeur du W3C, qui supervise l'élaboration des propositions de standards du Web sémantique.

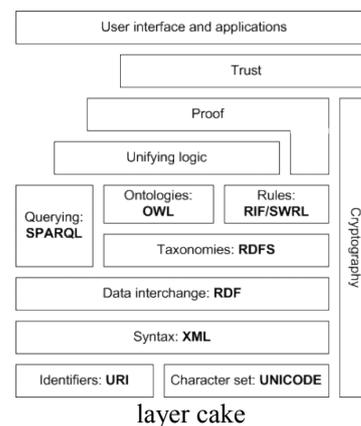
La plupart du temps, lorsque l'on prononce le terme de Web sémantique, on parle des différentes technologies qui se cachent derrière. Parmi les plus connues, on peut citer RDF (Ressource Description Framework) qui correspond à un modèle d'information, et les formats d'échanges de données en RDF pour communiquer entre différentes applications (RDF/XML, RDF/JSON, N3, Turtle, N-Triples et d'autres). Dans le domaine du Web sémantique, la sémantique des données est décrite par des ontologies - ce terme sera défini plus loin dans l'article - avec des langages prévus pour fournir une description formelle de concepts, termes ou relations d'un domaine quelconque. Ces langages sont RDFS (Ressource Description Framework Schema) et OWL (Web Ontology Language). Il existe aussi des langages de description des données structurées dans du XHTML afin que des outils effectuent un traitement automatique de ces différentes données. Ces langages sont RDFa et Microformat et, nouvellement arrivé avec HTML 5, Microdata. Voici d'ailleurs un article qui vous introduit le langage RDFa ([Lien 202](#)) et un autre sur les Microdata ([Lien 203](#)). Ensuite, pour finir avec la liste des technologies, il existe un langage de requête, au même titre que SQL pour les bases de données relationnelles, SPARQL, qui effectue des requêtes mais sur des triplets RDF. Il en existe d'autres (RQL et RDQL), mais ils sont bien moins utilisés.

2. Histoire du Web sémantique

En 1994, lors de la première conférence WWW à Genève, plus précisément au CERN, a lieu l'annonce de la création du W3C. C'est d'ailleurs à cette période que Tim Berners-Lee dresse les objectifs du W3C et montre les besoins d'ajouter de la sémantique au Web futur. Il montre alors en quoi les liens hypertextes ou, plus précisément, la façon dont on met en relation les documents sur le Web est trop limitée pour permettre aux machines de relier automatiquement les données contenues sur le Web à la

réalité. Compte tenu de l'ambition d'un tel projet, cette idée suscite quelques résistances et controverses qui sont classiquement rencontrées dès qu'on aborde des problématiques liées au domaine de l'intelligence artificielle.

Après cette conférence, mise à part la mise en place des recommandations nécessaires à la construction des documents, le W3C nouvellement créé entame les premières réflexions sur la mise en place du Web sémantique. Ces réflexions aboutissent à la publication d'un premier *draft* de recommandations sur le Web sémantique en octobre 1997 et d'une seconde en avril 1998. Cette même année, Tim Berners-Lee publie un document sur les toutes premières moutures de ce qui sera plus tard appelé le Web sémantique. Ces moutures consistent à mettre en place les différentes technologies du Web sémantique. Dans ce document, il présente le Web sémantique comme une sorte d'extension du Web des documents, qui constitue une base de données à l'échelle mondiale, afin que toutes les machines puissent mieux lier les données du Web. Cette feuille de route se matérialise par une représentation graphique, le « layer cake », qui montre l'agencement des différentes briques technologiques du Web sémantique.



Par ailleurs, en 1999, Tim Berners-Lee publie le livre *Weaving the Web* dans lequel il dresse un portrait du Web et les pistes pour son avenir. Les idées du Web sémantique n'en sont évidemment pas absentes. C'est d'ailleurs cette même année qu'il énonça sa célèbre citation :

I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers. A "Semantic Web", which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by

machines talking to machines. The "intelligent agents" people have touted for ages will finally materialize.

J'ai fait un rêve pour le Web [dans lequel les ordinateurs] deviennent capables d'analyser toutes les données sur le Web - le contenu, les liens et les transactions entre les personnes et les ordinateurs. Un « Web sémantique », qui devrait rendre cela possible, n'est pas encore sorti, mais, quand ce sera le cas, les mécanismes d'échange au jour le jour, la bureaucratie et notre vie quotidienne seront traités par des machines qui parlent à d'autres machines. Certains nous ont vanté depuis des lustres les « agents intelligents » et cela va enfin se concrétiser.

Citation provenant de : [Lien 204](#)

3. Objectifs du Web sémantique

Un des principaux objectifs du Web sémantique est de permettre aux utilisateurs d'utiliser la totalité du potentiel du Web : ainsi, ils pourront trouver, partager et combiner des informations plus facilement. Aujourd'hui tout le monde est capable d'utiliser des forums, d'utiliser des réseaux sociaux, de chatter, de faire des recherches ou même d'acheter différents produits. Néanmoins, il serait mieux que la machine fasse tout ceci à la place de l'homme, car actuellement, les machines ont besoin de l'homme pour effectuer ces tâches. La raison principale est que les pages Web actuelles sont conçues pour être lisibles par des êtres humains et non par des machines. Le Web sémantique a donc comme principal objectif que ces mêmes machines puissent réaliser seules toutes les tâches fastidieuses comme la recherche ou l'association d'informations et d'agir sur le Web lui-même.

4. Le Web sémantique et le Web 3.0

La communauté du Web dans son ensemble a tendance à dire que les deux termes " Web sémantique " et " Web 3.0 " représentent à peu près le même concept, si ce n'est pas totalement interchangeable. La définition continue de varier en fonction des gens avec qui vous parlez. L'avis général est que le Web 3.0 est très certainement la prochaine grande révolution, mais il se trouve que, pour le moment, ce ne sont que des spéculations quant à ce qu'il pourrait être. Il y aura encore de grosses améliorations, mais en gardant la plupart des propriétés du Web 2.0. Il y en a certains qui prétendent que le Web 3.0 sera plus applicatif et centrera ses efforts vers des environnements plus graphiques, d'autres qui prétendent qu'il sera plus axé sur la recherche d'information géographique basées sur la géolocalisation ou encore même utiliser les nombreux progrès en intelligence artificielle.

5. Le Web sémantique souvent critiqué

Comme vous avez pu le deviner, le Web sémantique permet à tous d'en savoir plus sur les sujets qui les intéressent. Cela signifie *a contrario* que diverses institutions peuvent récolter des informations sur vous librement et donc constituer des dossiers. Ces institutions peuvent être des agences de publicité, de sécurité ou même des services de renseignements. Et ceci tout en restant dans la légalité car, dans 90 % des cas, ces informations sont mises en ligne volontairement ou non par les

utilisateurs, sans qu'ils se doutent du « danger » que peut représenter cette action.

Le Web sémantique est aussi critiqué à cause de sa lourdeur : les langages utilisés pour le Web sémantique sont très verbeux car dérivés du XML et donc souvent pénible à utiliser. De ce fait, l'écriture d'ontologies est souvent très problématique car elle exige une spécialisation dans un domaine particulier et, lorsque l'on ne maîtrise pas ce domaine, elle devient très difficile à créer. Ainsi, certaines personnes disent qu'il est préférable d'utiliser des « words tags » (ce sont une série de mots clés qui permettent de qualifier une ressource) à la place des ontologies.

6. L'ontologie, outil principal du Web sémantique

Nous allons dans cette partie expliquer de manière non technique ce qu'est une ontologie.

6.1. Définition

L'ontologie est la base de ce que l'on appelle la représentation des connaissances. Ce domaine est né de la volonté des chercheurs de représenter diverses connaissances du monde actuel, de façon à ce qu'elles soient utilisables par des ordinateurs, pour qu'ils puissent effectuer des raisonnements sur ces connaissances. Ces connaissances sont exprimées sous forme de symboles auxquels on donne une « sémantique » (un sens).

Imaginons la problématique suivante : vous voulez interroger une base de données contenant diverses ressources (textes, images, vidéos...) et une requête (question ou mot(s) clé(s)), comment trouver les données se trouvant dans cette base qui correspondent à cette requête ?

Par exemple tapez dans votre moteur de recherche préféré les mots suivants : « ordinateur portable » puis « laptop ». Vous pouvez vous apercevoir que les résultats ne sont pas du tout les mêmes, alors que, vu que les deux mots représentent la même chose, on pourrait s'attendre à trouver les mêmes réponses.

Que se passe-t-il ? En fait, le moteur de recherche compare des mots sans prendre en compte leur sémantique (sens). Il exécute uniquement une recherche strictement syntaxique et donc sans réflexion car « ordinateur portable » et « laptop » représentent le même concept (la même chose), que nous appellerons maintenant des classes pour respecter la terminologie du Web sémantique. Plus précisément, on peut dire que la relation de spécialisation sur les classes n'est pas gérée. Par exemple, « notebook » est une spécialisation de la classe générale « laptop ». Ainsi, pour raisonner, il ne faut plus se baser sur les mots mais sur les classes. Mais que signifie raisonner ? Raisonner c'est utiliser sa raison pour démontrer quelque chose. C'est un terme très souvent employé en intelligence artificielle.

Illustration en comparant deux moteurs de recherche :

Avec Orange :



Avec Google :



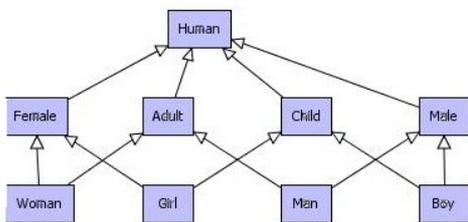
Vous pouvez donc voir que sur le moteur d'Orange, la première occurrence est la réponse à notre question et qu'ensuite il y a les réponses résultant de la recherche syntaxique. Alors que sur Google il n'y a que la recherche syntaxique. Cette différence est tout simplement due à l'utilisation d'ontologies de la part du moteur d'Orange.

Ainsi pour résoudre ce problème on construit ce que l'on appelle des bases de connaissances qui sont constituées :

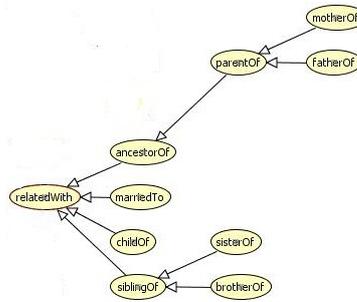
- d'une **ontologie** : une collection de classes et de relations (que nous appellerons propriété pour les mêmes raisons que concept) entre ces classes ;
- de **règles** : une expression de contraintes sur les propriétés et les classes de l'ontologie ;
- de **faits** : des instances de l'ontologie.

6.2. Exemple de construction d'une ontologie

Faisons maintenant un petit exemple d'ontologie sur la famille Simpson pour illustrer tout ceci. Tout d'abord les concepts : {Human, Child, Boy, Girl, Male, Man, Female, Woman, Adult} qui nous donne la représentation suivante entre ces classes :



Ensuite, occupons-nous des propriétés : {relatedWith(Human, Human), siblingOf(Human, Human), sisterOf(Female, Human), brotherOf(Male, Human), ancestorOf(Adult, Human), parentOf(Adult, Human), fatherOf(Man, Human), motherOf(Woman, Human), marriedTo(Adult, Adult), childOf(Human, Adult)}. Ce qui nous donne la représentation suivante entre ces propriétés :



Maintenant admettons quelques règles, logiques pour un humain sur ces classes et propriétés :

- règle 1 : si une classe "Male" appartient à une propriété "brotherOf" avec une classe "Female" alors la classe "Female" appartient à la propriété "sisterOf" avec la classe "Male" ;
- règle 2 : réciproque de la règle 1 ;
- règle 3 : si une classe "Adult" appartient à une propriété "parentOf" avec une classe "Human" alors la classe "Human" appartient à la propriété "childOf" avec la classe "Adult" ;
- règle 4 : réciproque de la règle 3.

De cette manière, on peut facilement imaginer toutes les règles possibles entre ces propriétés et classes. Pour finir prenons quelques instances : {Bart, Lisa, Homer}. On dit ensuite que l'instance "Bart" appartient à la classe "Boy", "Lisa" appartient à la classe "Girl" et "Homer" appartient à la classe "Man". On dit ensuite que "Bart" est le frère de "Lisa" et que "Bart" et "Lisa" sont les enfants de "Homer" avec nos propriétés.

Notre ontologie ainsi construite, il nous est maintenant facile d'effectuer les requêtes suivantes dessus :

- est-ce que Lisa est la sœur de Bart ?
- est-ce que Homer est un parent de Lisa et Bart ?

Bien entendu cette ontologie est ridiculement petite et facile comparée aux ontologies de domaines plus complexes. Mais, maintenant que vous savez ce qu'est une ontologie, vous pouvez vous amuser à continuer l'ontologie de la famille Simpson en ajoutant des propriétés, des classes, des règles et des instances.

6.3. Quelques ontologies déjà existantes

Je vous donne ici une liste non exhaustive des ontologies existantes :

- FOAF : [Lien 205](#) ;
- EVENT : [Lien 206](#) ;
- DC : [Lien 207](#) ;
- OAN : [Lien 208](#) ;
- D2R : [Lien 209](#) ;
- VCARD : [Lien 210](#) ;
- CREATIVE COMMONS : [Lien 211](#).

7. Conclusion

Vous en savez maintenant beaucoup plus sur le Web sémantique et j'espère que je vous ai donné envie d'en apprendre plus sur ce domaine encore trop méconnu.

Retrouvez l'article de Julien Plu en ligne : [Lien 212](#)

Introduction au RDF et à l'API RDF de Jena

Ceci est un tutoriel introductif à la fois au framework de description de ressources (RDF, Resource Description Framework) du W3C et à Jena, une API Java pour RDF. Il est écrit pour le développeur peu familier à RDF et qui apprend le mieux en prototypant ou, pour d'autres raisons, désire aller rapidement à l'implémentation. Une certaine habitude à XML et à Java est requise.

Implémenter trop vite, sans d'abord comprendre le modèle de données de RDF, mène à la frustration, à la déception. Cependant, étudier le modèle de données seul est assez difficile et mène généralement à des énigmes métaphysiques tortueuses. Il vaut mieux approcher la compréhension du modèle de données et de son utilisation en parallèle, apprendre un peu du modèle et l'essayer, puis passer au morceau suivant. Ainsi, la théorie complète la pratique et la pratique complète la théorie. Le modèle de données est assez simple, cette approche ne devrait pas prendre trop de temps.

RDF possède une syntaxe XML et tous ceux qui sont familiers avec XML le prendront pour tel. C'est une erreur. RDF devrait être compris en termes de son modèle de données. Des données RDF peuvent être représentées en XML, mais comprendre la syntaxe est secondaire à celle du modèle de données.

Une implémentation de l'API Jena, incluant le code source de travail pour tous les exemples de ce tutoriel, peut être téléchargée depuis le site officiel de Jena : [Lien 213](#).

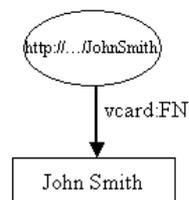
1. L'article original

Cet article est la traduction de An Introduction to RDF and the Jena RDF API : [Lien 214](#).

2. Introduction

RDF est un standard (d'un point de vue purement technique, une recommandation du W3C) pour la description de ressources. Qu'est-ce qu'une ressource ? C'est une question plutôt profonde et la définition précise est toujours l'objet de tumultueux débats. Pour notre utilisation, on peut imaginer qu'une ressource est quelque chose que l'on peut identifier. Vous êtes une ressource, au même titre que votre page d'accueil, ce tutoriel ou la baleine blanche de Moby Dick.

Nos exemples seront des personnes. Ils utilisent une représentation RDF des VCARDS : [Lien 215](#). RDF se représente le mieux sous la forme de nœuds et de diagrammes en arcs. Une VCARD simple pourrait ressembler à ceci en RDF :



VCARD en RDF

La ressource, John Smith, est représentée comme une ellipse et est identifiée par un identifiant de ressource uniforme (URI, Uniform Resource Identifiant), dans ce cas `http://.../JohnSmith`. Si vous essayez d'accéder à cette ressource dans votre navigateur, vous n'y arriverez probablement pas ; résistant à la tentation du poisson d'avril, il serait étonnant qu'un navigateur puisse vous amener John Smith. Si vous n'êtes pas familier aux URI, prenez-les simplement comme des noms d'apparence assez

bizarre.

Les ressources possèdent des propriétés. Dans ces exemples, on s'intéresse au type de propriétés qui pourraient apparaître sur la carte de visite de ce John Smith. La première figure montre une propriété, son nom complet. Une propriété est représentée par un arc, nommé avec la propriété. Le nom de la propriété est aussi une URI, mais une URI plutôt longue et encombrante, le diagramme montre donc une forme XML. La partie avant : s'appelle le préfixe d'espace de noms et représente un espace de noms. La partie après : est appelée un nom local et représente un nom dans cet espace. Les propriétés sont habituellement représentées dans cette forme *qname* dans du RDF XML et c'est un raccourci pratique pour les représenter dans des diagrammes et du texte. Strictement parlant, cependant, les propriétés sont identifiées par une URI. La forme `nsprefix:localname` est un raccourci pour cette URI de l'espace de noms concaténé au nom local. Il n'y a pas besoin que l'URI de la propriété donne accès à quelque chose par le biais d'un navigateur.

Chaque propriété a une valeur. Dans ce cas, la valeur est littérale, soit, pour le moment, une chaîne de caractères. Ces littérales sont montrées dans des rectangles.

Jena est une API Java qui peut être utilisée pour créer et manipuler des graphes RDF comme celui-ci. Jena possède des classes pour représenter des graphes, des ressources, des propriétés et des littérales. Les interfaces représentant des ressources, des propriétés et des littérales sont respectivement nommées `Resource`, `Property` et `Literal`. Dans Jena, un graphe est appelé un modèle et est représenté par l'interface `Model`.

Le code pour créer ce graphe (ou modèle) est simple :

```
static String personURI =
    "http://somewhere/JohnSmith";
static String fullName = "John Smith";
Model model = ModelFactory.createDefaultModel();
```

```
Resource johnSmith =
model.createResource(personURI);
johnSmith.addProperty(VCARD.FN, fullName);
```

Il commence avec des définitions de constantes et ensuite crée un modèle vide en utilisant la méthode `createDefaultModel()` de `ModelFactory` pour créer un modèle en mémoire. Jena contient d'autres implémentations de l'interface `Model`, par exemple celle qui utilise une base de données relationnelle (SGBDR) : ces types de modèles sont aussi disponibles depuis `ModelFactory`.

La ressource John Smith est alors créée et une propriété y est ajoutée. La propriété est fournie par une classe « constante » `VCARD`, qui détient les objets représentant toutes les définitions dans le schéma `VCARD`. Jena fournit des classes constantes pour d'autres schémas bien connus, comme `RDF` et `RDF Schema` eux-mêmes, `Dublin Core` et `DAML`.

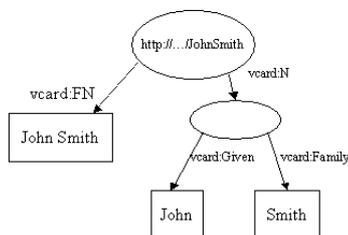
Le code pour créer la ressource et ajouter la propriété peut être écrit d'une manière plus compacte avec un style en cascade :

```
Resource johnSmith =
    model.createResource(personURI)
        .addProperty(VCARD.FN, fullName);
```

Le code de travail pour cet exemple peut être trouvé dans le répertoire `/src-examples` de la distribution Jena, dans le fichier `Tutorial01` : [Lien 216](#). Comme exercice, prenez ce code et modifiez-le pour créer une `VCARD` simple vous-même.

Maintenant, ajoutons un peu de détails à cette `VCARD`, en explorant d'autres fonctionnalités de Jena et de `RDF`.

Dans ce premier exemple, la valeur de la propriété était littérale ; les propriétés `RDF` peuvent aussi prendre d'autres ressources comme valeur. En utilisant une technique `RDF` courante, cet exemple montre comment représenter les différentes parties du nom de John Smith :



Ici, on a ajouté une nouvelle propriété, `vcard:N`, pour représenter la structure du nom de John Smith. Il y a plusieurs choses dignes d'intérêt dans ce modèle. Notez que la propriété `vcard:N` prend une ressource comme valeur. Notez aussi que l'ellipse représentant le nom composé n'a pas d'URI. On appelle cela un nœud anonyme.

Le code Jena pour construire cet exemple est encore une fois très simple. Quelques déclarations et puis la création du modèle vide.

```
String personURI = "http://somewhere/JohnSmith";
String givenName = "John";
String familyName = "Smith";
String fullName = givenName + " " + familyName;
Model model = ModelFactory.createDefaultModel();
Resource johnSmith
    = model.createResource(personURI)
        .addProperty(VCARD.FN, fullName)
        .addProperty(VCARD.N,
            model.createResource()
                .addProperty(VCARD.Given,
                    givenName)
                .addProperty(VCARD.Family,
                    familyName));
```

Le code de travail de cet exemple peut être trouvé comme `tutorial-2` dans le même répertoire de la distribution Jena : [Lien 217](#).

3. Déclaration (statement)

Chaque arc dans un modèle `RDF` est appelé une déclaration. Chacune affirme un fait à propos d'une ressource. Une déclaration se décompose en trois parties :

- le **sujet** est la ressource que quitte l'arc ;
- le **prédicat** est la propriété qui donne un nom à l'arc ;
- l'**objet** est la ressource ou la valeur littérale pointée par l'arc.

Une telle déclaration est donc parfois appelée un triplet, à cause de ces trois parties.

Un modèle `RDF` est représenté comme un ensemble de ces déclarations. Chaque appel à `addProperty()` ajoute une autre déclaration au modèle. Puisqu'un modèle est un ensemble de déclarations, ajouter une déclaration en double n'a pas d'effet. Les interfaces de modèle de Jena définissent une méthode `listStatements()`, qui retourne un `StmtIterator`, un dérivé de la classe Java `Iterator` sur toutes les déclarations dans un modèle. `StmtIterator` possède une méthode `nextStatement()`, qui retourne la déclaration suivante de l'itérateur (le même que `next()` retournerait, déjà casté en `Statement`). L'interface `Statement` fournit des méthodes d'accès au sujet, au prédicat et à l'objet de la déclaration.

Maintenant, on va utiliser cette interface pour étendre le deuxième exemple, lister toutes les déclarations créées et les afficher. Le code complet se trouve dans `tutorial 3` : [Lien 218](#).

```
StmtIterator iter = model.listStatements(); while
(iter.hasNext()) {
    Statement stmt = iter.nextStatement(); // get
next statement
    Resource subject = stmt.getSubject(); // get
the subject
    Property predicate = stmt.getPredicate(); //
get the predicate
    RDFNode object = stmt.getObject(); // get
the object System.out.print(subject.toString());
System.out.print(" " + predicate.toString() +
");
    if (object instanceof Resource) {
        System.out.print(object.toString());
    } else {
        // object is a literal
    }
}
```

```
System.out.print(" \"" + object.toString() +
"\");
} System.out.println(" .");
}
```

Puisque l'objet d'une déclaration peut être une ressource ou une valeur littérale, la méthode getObject() retourne un objet de type RDFNode, une superclasse commune à Resource et à Literal. L'objet sous-jacent est du type approprié, ainsi le code utilise instanceof pour déterminer lequel et le traiter en fonction.

Quand il est lancé, ce programme devrait produire une sortie comme ceci :

```
http://somewhere/JohnSmith
http://www.w3.org/2001/vcard-rdf/3.0#N
anon:14df86:ecc3dee17b:-7fff .
anon:14df86:ecc3dee17b:-7fff
http://www.w3.org/2001/vcard-rdf/3.0#Family
"Smith" .
anon:14df86:ecc3dee17b:-7fff
http://www.w3.org/2001/vcard-rdf/3.0#Given "John"
.
http://somewhere/JohnSmith
```

```
http://www.w3.org/2001/vcard-rdf/3.0#FN "John
Smith" .
```

Maintenant, vous savez pourquoi il est plus simple de dessiner des modèles. En y regardant de plus près, on peut voir que chaque ligne consiste en trois champs, représentant le sujet, le prédicat et l'objet de chaque déclaration. Il y a quatre arcs dans le modèle, donc quatre déclarations. anon:14df86:ecc3dee17b:-7fff est un identifiant interne généré par Jena. Il ne s'agit pas d'une URI et il ne devrait pas être confondu avec une URI. Il s'agit simplement d'une dénomination interne utilisée par l'implémentation de Jena.

Le groupe de travail du W3C sur RDFCore ([Lien 219](#)) a défini une notation simple et similaire, les N-triplets : [Lien 220](#). Le nom signifie « triple notation ». On verra dans la prochaine section que Jena a un module d'écriture de N-triplets incorporé.

Retrouvez la suite de l'article de Brian McBride, Daniel Boothby et Chris Dollin traduit par Thibaut Cuvelier et Julien Plu en ligne : [Lien 221](#)

Initiation au décisionnel (Business Intelligence, DataWarehouse, OLAP)

Ce tutoriel a pour objectif de vous expliquer les objectifs du décisionnel ainsi que le jargon qui va avec. Dans une deuxième partie, je passe à un tutoriel qui traite l'analyse OLAP qui est partie intégrante du décisionnel.

1. Introduction

Vous avez certainement dû entendre parler d'au moins d'un de ces trois termes qui sont intimement liés : Business intelligence, Datawarehouse et Analyse OLAP. En effet, depuis les années 2000-2001, le marché du décisionnel ne cesse d'exploser en France (et partout d'ailleurs) surtout avec l'investissement de plusieurs grandes sociétés qui souhaitent instaurer un système de Business Intelligence (B.I.) dans leur organisation. Ce système difficile à mettre en œuvre, demandant une expertise et nécessitant une maîtrise d'ouvrage de la part des informaticiens concepteurs du système rend souvent difficile le recrutement de ces profils. Mais c'est quoi le décisionnel ? Comment y débiter ? Eh bien je vais tenter dans cet article de donner un aperçu de tout ce jargon souvent méconnu et n'ayant souvent pas d'équivalents français et j'essaierai après un bref aspect théorique, de passer à un aspect pratique traitant l'analyse OLAP avec Analysis Services de Microsoft.

NB : la deuxième partie de cet article a été réalisée avec la version 7.0 de SQL Server. Étant encore nouveau dans la version 2005, j'attends encore un moment pour réaliser une nouvelle version de cette partie pratique avec SQL Server 2005.

2. Aspect théorique

Dans cet aspect théorique nous allons un peu expliquer l'utilité du décisionnel, des acteurs du décisionnel et des architectures usuelles.

2.1. Pourquoi le décisionnel ?

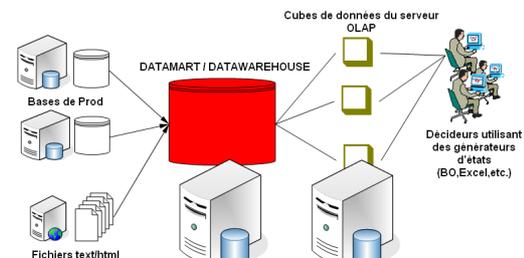
Tout d'abord, rappelons-le, le décisionnel ne concerne souvent que les entreprises qui gèrent un historique de leurs événements passés (faits, transactions, etc.). Les entreprises qui viennent de naître n'ont souvent pas besoin de faire du décisionnel car elles n'ont pas encore besoin de catégoriser ou de fidéliser leurs clients. Le souci majeur pour elles serait plutôt d'avoir le maximum de clients et c'est après en avoir récupéré un grand nombre qu'elles penseront certainement à les fidéliser et leur proposer d'autres produits susceptibles de les intéresser. C'est ce que l'on appelle Customer RelationShip Management (CRM ou gestion des relations clients).

2.2. Qui a besoin du décisionnel ?

Comme cela peut se deviner, les décideurs sont les principaux utilisateurs des systèmes décisionnels. Les décideurs sont généralement des « marketeurs » ou analystes en général. Ces derniers établissent généralement des plans marketing qui leur permettent de mieux cibler

leurs clients, de les fidéliser, etc. Et pour cela, ils ont besoin d'indicateurs et des données résumées de leurs activités (ils n'ont souvent besoin de détail que pour des cas spécifiques). Par exemple, contrairement aux systèmes relationnels (ou base gestion) où les utilisateurs chercheront à connaître leurs transactions pour faire un bilan, les systèmes décisionnels quant à eux cherchent plutôt à donner un aperçu global pour connaître les tendances des clients (d'où l'opposition des deux modes [quantitatif contre qualitatif]).

2.3. Architecture des systèmes décisionnels



Exemple d'architecture décisionnelle

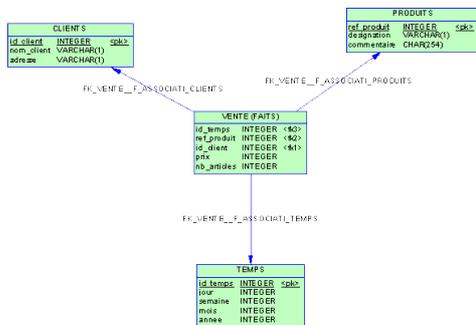
Voici une architecture de système décisionnel très utilisée. Dans cette architecture, on dispose d'un entrepôt de données ou DataWarehouse (généralement, il s'agit plutôt d'un datamart qui est plus petit que le DW et qui concerne un domaine bien particulier [finance, ressources humaines, etc.]). L'entrepôt (ou encore info-centre !) centralise les données issues de plusieurs sources (bases de production de l'entreprise, fichiers textes, documents web [html, xml, sgml, etc.] etc.). Ces données sont fusionnées dans l'entrepôt qui est généralement une grosse base de données (SQL Server, Oracle, etc.). Ensuite, une fois l'entrepôt confectionné, des données sont extraites dans des serveurs d'analyse ou serveurs OLAP sous forme de cubes de données (Analysis Server, EssBase, etc.) afin d'être analysées. Enfin, des générateurs d'états (Business Objects, Crystal Report, etc.) sont utilisés afin de présenter l'étude aux utilisateurs finaux ou décideurs (ex. : analystes marketing).

2.3.1. Les sources de données

Les sources de données sont souvent diverses et variées et le but est de trouver des outils ETL (Extraction / Transformation / Loading) afin de les extraire, de les nettoyer, de les transformer et de les mettre dans l'entrepôt de données (DTS de SQL Server est un exemple d'outil ETL.). Des outils comme Datastage ou Talend (monde open source) sont spécialisés en la matière.

2.3.2. L'entrepôt de données

Il est le cœur du système décisionnel et demande une analyse profonde de la part de la maîtrise d'ouvrage. La conception d'un DataWarehouse diffère de la conception d'une base de données relationnelle. En effet, alors que les bases de données relationnelles tendent le plus souvent à être normalisées, les bases de données multidimensionnelles, elles, sont plutôt dénormalisées, respectant le modèle en étoile ou le modèle en flocon. Voici ci-dessous un exemple de schéma d'un entrepôt de données :



Modèle physique d'un schéma en étoile.

Le modèle physique ci-dessus contient une table centrale à laquelle toutes les autres tables sont liées (modèle en étoile). La table centrale (ici table VENTE) est appelée la table des faits et contient toutes les autres clés des autres tables. Cette table de faits contient aussi une ou plusieurs valeurs numériques particulières (ici prix et nb_articles) appelées mesures. Généralement un niveau de granularité est aussi défini pour la table des faits (regroupe-t-on par exemple un ensemble de ventes de même type pour en faire un enregistrement ? Les enregistrements sont-ils unitaires (un enregistrement par transaction ?). Les autres tables du modèle sont appelées tables de dimensions. Ici par exemple, on dispose des dimensions CLIENTS, PRODUITS et TEMPS. Une dimension Temps est presque toujours présente dans les bases multidimensionnelles tout simplement parce qu'on analyse les données dans le temps.

2.3.3. Le serveur OLAP ou serveur d'analyse

OLAP (On-Line Analytical Processing) est opposé à OLTP (On-Line Transactional Processing) et a pour but d'organiser les données à analyser par domaine/thème et d'en ressortir des résultats pertinents pour le décideur. Les résultats sont donc des résumés et peuvent être obtenus par différents algorithmes de datamining (fouille de données) du serveur d'analyse. On peut par exemple établir le résultat suivant : « Les clients qui achètent généralement du beurre et du pain achètent aussi du lait ». Ces résultats pourraient amener l'organisation (ici en l'occurrence une grande distribution) à disposer ses rayons de telle façon qu'à côté de l'emplacement du beurre, elle mettra le pain et le lait.

2.3.4. Le générateur d'états

Le générateur d'état permet seulement de mieux appréhender le résultat de l'analyse. L'utilisateur final n'étant pas forcément un informaticien, il aura plus de facilité dans des états Business Objects (ou même dans des feuilles de données Excel) avec des diagrammes et courbes statistiques que d'aller directement requêter dans le serveur

d'analyse. Au passage, je rappelle que travaillant généralement avec SQL Server et Analysis Services, le langage de requêtage multidimensionnel a pour nom MDX (qui ressemble au SQL mais n'est pas du SQL). Les états permettent également de faire de l'exploration (navigation) de données (notamment du Rollup / Drill-Down).

2.3.5. Quelques termes usuels du décisionnel

datawarehouse : entrepôt de données.

datamart : petit entrepôt de données à l'échelle d'un département ou succursale d'une grande société. Généralement un datamart déverse ses données chez sa mère qui est le datawarehouse.

OLTP : OnLine Transactional Processing. Il s'agit des traitements transactionnels. Par exemple, les logiciels des caisses enregistreuses des chaînes de magasins font du OLTP.

OLAP : OnLine Analytical Processing. Opposé à l'OLTP, faire de l'OLAP signifie faire de l'analyse de données. Analyser les ventes, détecter les fraudes, prospecter des clients font partie du processus OLAP.

ETL : un outil ETL (Extraction / Transformation / Loading) permet à partir de diverses sources de données, d'extraire de l'information, de faire des transformations afin de nettoyer les données et de charger des données utiles dans l'entrepôt de données. Les sources de données peuvent être diverses (HTML, XML, base de données, fichiers texte, tableurs, ERP, etc.).

Serveur d'analyse : un serveur d'analyse ou serveur OLAP est un serveur de base de données multidimensionnelle. Exemple : Analysis Server est un serveur de bases multidimensionnelles.

Base de données multidimensionnelle : une base de données multidimensionnelle par opposition à une base de données relationnelle est une base dénormalisée ou il existe une table centrale (table de faits) liée à toutes les autres tables (tables de dimensions).

Table de faits : comme son nom l'indique, une table de faits est une table contenant tous les faits du SI et dont dépendent toutes les autres tables. Cette table ne contient que des clés étrangères venant des tables de dimensions et des valeurs numériques appelées mesures. Exemple de table de faits : table des ventes

Tables de dimensions : les tables de dimensions sont des tables servant d'axes d'analyse. On peut par exemple analyser les ventes (table de faits) suivant l'axe des temps (table de dimensions) pour indiquer par exemple pendant quel trimestre de l'année les ventes ont explosé.

Mesure : une mesure est une quantité présente dans la table de faits qui permet de mesurer les faits. Par exemple, nombre de ventes ou prix unitaire sont des exemples de mesures.

Cube : un cube de données est une structure dimensionnelle comme une table est une structure

relationnelle. Un cube est constitué d'une ou plusieurs tables de faits avec leurs tables de dimensions. On peut par exemple considérer un cube vente contenant sa table de faits « vente » et ses tables de dimensions « clients », « région » et « temps ».

Niveau de hiérarchie : un niveau de hiérarchie se définit au niveau des tables de dimensions. Cela permet d'agréger les données. Par exemple, supposons qu'on ait la dimension région contenant la liste des villes, on pourrait faire un niveau de hiérarchie (niveau 1) classant les villes en régions, ensuite un niveau plus bas qui les classerait en départements (niveau 2).

Drill-down: faire un drill-down, c'est avoir un niveau de détails sur les données. Par exemple Supposons qu'on veuille voir le détail des ventes pour le premier trimestre de l'année 1997. On dit qu'on fait un drill-down sur l'axe (ou dimension) temps. C'est-à-dire qu'on ne veut pas voir seulement les données de l'année 1997 mais descendre à un niveau de détail plus bas.

Roll-up: rollup est le contraire de drill-down. C'est donc faire de l'agrégation (ou résumé) des données.

Retrouvez l'article de Mohamed Taslimanka Sylla en ligne : [Lien 222](#)

Enterprise Resource Planning ou Progiciel de Gestion Intégré

Émanant d'un concepteur unique, un ERP est un progiciel qui permet de gérer l'ensemble des processus d'une entreprise intégrant l'ensemble de ses fonctions comme la gestion des ressources humaines, la gestion financière et comptable, l'aide à la décision, la vente, la distribution, l'approvisionnement, la production ou encore du e-commerce.

Le principe fondateur d'un ERP est de construire des applications informatiques correspondant aux diverses fonctions citées précédemment de manière modulaire sachant que ces modules sont indépendants entre eux, tout en partageant une base de données unique et commune au sens logique.

L'autre principe qui caractérise un ERP est l'usage de ce qu'on appelle un moteur de workflow et qui permet, lorsqu'une donnée est enregistrée dans le SI, de la propager dans les modules qui en ont l'utilité, selon une programmation prédéfinie.

Ainsi, on peut parler d'ERP lorsqu'on est en présence d'un SI composé de plusieurs applications partageant une seule et même base de données, par le biais d'un système automatisé prédéfini et éventuellement paramétrable, un moteur de workflow.

Il existe des ERP propriétaires et Open Source. Pour en citer quelques-uns :

ERP Propriétaires

- SAP
- Oracle/PeopleSoft
- Navision de Microsoft
- Sage Adonix
- Cegid

Open Source

- Compiere
- ERP5
- OpenERP
- Dolibarr
- OpenBravo

Forums ERP/PGI : [Lien 223](#)

Tutoriels ERP/PGI : [Lien 224](#)

Customer Relationship Management ou Gestion Relation Client

Un CRM, soit Customer Relationship Management ou encore GRC pour Gestion de la Relation Client, est un outil révolutionnaire dans le domaine du marketing. En effet, un CRM dépasse l'approche marketing traditionnelle

focalisée sur la relation entre un produit/service et son client ou encore la notoriété et l'image de marque. Grâce aux nombreuses fonctionnalités offertes par le CRM telles que la gestion des prospects, gestion des clients et de leurs achats, de reporting... Il est possible pour l'entreprise de personnaliser l'offre, sous forme de publicités ou encore d'e-mails afin de fidéliser un maximum le client, d'améliorer leur satisfaction et par la même occasion d'accroître l'efficacité des employés et par conséquent d'augmenter les ventes.

Quelques exemples d'outils CRM :

- Microsoft CRM
- SalesForce
- Oracle/PeopleSoft crm
- Sage crm
- SAP CRM
- Selligent
- Siebel
- ...

Tutoriels CRM/GRC : [Lien 225](#)

Business Intelligence ou informatique décisionnelle

Partie de l'informatique consistant à effectuer du reporting automatisé et des analyses numériques à partir de données existantes.

Idéalement, les données sont extraites par un ETL (Extract, Transform and Load) des bases de production ou autres sources (fichiers plats, tableurs, etc.) et intégrées dans un Datawarehouse, agrégées dans un ou plusieurs Datamart. Elles sont lues soit directement par un outil de reporting, soit à travers une couche sémantique pour une analyse ad hoc.

Les limites de la BI ne sont pas clairement définies. Si l'ensemble des éléments cités ci-dessus en font clairement partie, d'autres types d'outils logiciels (le Performance Management, la consolidation statutaire, l'élaboration budgétaire et autres outils financiers, le datamining, la segmentation et autres outils statistiques) sont parfois considérés comme de la BI et parfois comme relevant de domaines autonomes. Mais de plus en plus, la tendance est à la concentration des éditeurs et à une convergence de ces outils.

Forums Business Intelligence : [Lien 226](#)

Tutoriels Business Intelligence : [Lien 227](#)

Electronic Document Management ou Gestion électronique des documents

La gestion électronique de document est un système permettant l'intégration, l'organisation et la hiérarchisation,

le stockage et l'archivage de documents en particulier dès que le traitement est appliqué sur d'importants volumes documentaires. Tout document numérique peut être géré par un système de GED (courrier électronique, pdf, word, images, etc.).

Ce type de système est utilisé pour de gros volumes documentaires mais aussi afin de faciliter le travail collaboratif (par exemple lors de validation documentaire) au sein d'une équipe ou d'une entreprise.

Quelques exemples d'outils ECM/GED :

- Nuxeo
- Oracle/PeopleSof

- Maarch
- KnowledgeTree
- Freedom
- IBM (DB2 Content Manager, Content Manager OnDemand)
- Forrester
- EMC Documentum
- SyGED

Forums ECM/GED : [Lien 228](#)

Retrouvez ces définitions sur la rubrique Solutions d'entreprise : [Lien 229](#)

Introduction à la Programmation Orientée Objet

Dans ce tutoriel vous apprendrez à manier la Programmation Orientée Objet, ou POO pour les intimes, dans le cadre du langage Pascal Orienté Objet.

Loin d'être aussi complexe qu'elle peut le laisser transparaître, la POO peut se maîtriser rapidement au point de ne plus pouvoir s'en passer.

1. Avant-Propos

Si la programmation dite procédurale est constituée de procédures et fonctions sans liens particuliers agissant sur des données dissociées pouvant mener rapidement à des difficultés en cas de modification de la structure des données, la **programmation objet**, pour sa part, tourne autour d'une unique entité : l'*objet*, offrant de nouvelles perspectives, et que je vous invite à découvrir de suite...

Attention !

Borland a longtemps employé le nom de *Pascal Objet* pour **Delphi**. Celui-ci a été récemment renommé *langage Delphi*. Nous n'aborderons pas dans ce tutoriel une approche spécifique à **Delphi**. Nous nous orienterons plus vers une approche générale du Pascal, **tous compilateurs Pascal confondus** sitôt que ceux-ci supportent la Programmation Orientée Objet, comme c'est le cas pour **Turbo Pascal, Free Pascal, GNU Pascal...** et bien sûr **Delphi**.

2. Vue d'ensemble de la POO

Avant de rentrer plus avant dans le sujet qui nous intéresse, nous allons commencer par poser un certain nombre de bases.

2.1. L'objet

Il est impossible de parler de Programmation Orientée Objet sans parler d'*objet*, bien entendu. Tâchons donc de donner une définition aussi complète que possible d'un *objet*.

Un objet est avant tout une **structure de données**. Autrement, il s'agit d'une entité chargée de gérer des données, de les classer, et de les stocker sous une certaine forme. En cela, rien ne distingue un *objet* d'une quelconque autre structure de données. La principale différence vient du fait que l'*objet regroupe les données et les moyens de traitement de ces données*.

Un *objet* rassemble de fait deux éléments de la programmation procédurale.

- Les **champs** :

Les *champs* sont à l'objet ce que les variables sont à un programme : ce sont eux qui ont en charge les données à gérer. Tout comme n'importe quelle autre variable, un *champ* peut posséder un type quelconque défini au préalable : nombre, caractère... ou même un type objet.

- Les **méthodes** :

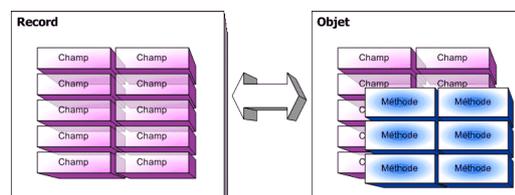
Les *méthodes* sont les éléments d'un objet qui servent d'interface entre les données et le programme. Sous ce nom obscur se cachent simplement des procédures ou fonctions destinées à traiter les données.

Les champs et les méthodes d'un objet sont ses **membres**.

Si nous résumons, un *objet* est donc un type servant à stocker des données dans des *champs* et à les gérer au travers des *méthodes*.

Si on se rapproche du Pascal, un objet n'est donc qu'une extension évoluée des *enregistrements* (type **record**) disposant de procédures et fonctions pour gérer les champs qu'il contient.

On notera souvent les membres d'un objet **Objet.Membre** de façon à lever toute ambiguïté quant au propriétaire du membre considéré.



2.2. Objet et classe

Avec la notion d'**objet**, il convient d'amener la notion de **classe**. Cette notion de **classe** n'est apparue dans le langage Pascal qu'avec l'avènement du langage Delphi et de sa nouvelle approche de la Programmation Orientée Objet. Elle est totalement absente du Pascal standard.

Ce que l'on a pu nommer jusqu'à présent *objet* est, pour Delphi, une **classe d'objet**. Il s'agit donc du type à proprement parler. L'**objet** en lui-même est une **instance de classe**, plus simplement un exemplaire d'une classe, sa représentation en mémoire.

Par conséquent, on déclare comme type une *classe*, et on déclare des variables de ce type appelées des *objets*.

Si cette distinction est à bien prendre en considération lors de la programmation en Delphi, elle peut toutefois être totalement ignorée avec la plupart des autres compilateurs Pascal. En effet, ceux-ci ne s'appuient que sur les notions d'**objet** et d'**instance d'objet**.

Nous adopterons par conséquent ici ce point de vue qui simplifie le vocabulaire et la compréhension.

On pourra remarquer que Free Pascal pour sa part définit une **classe** comme un "pointeur vers un objet ou un enregistrement".

2.3. Les trois fondamentaux de la POO

La Programmation Orientée Objet est dirigée par trois fondamentaux qu'il convient de toujours garder à l'esprit : **encapsulation**, **héritage** et **polymorphisme**. Houlà ! Inutile de fuir en voyant cela, car en fait, ils ne cachent que des choses relativement simples. Nous allons tenter de les expliquer tout de suite.

2.3.1. Encapsulation

Derrière ce terme se cache le concept même de l'objet : réunir sous la même entité les données et les moyens de les gérer, à savoir les champs et les méthodes.

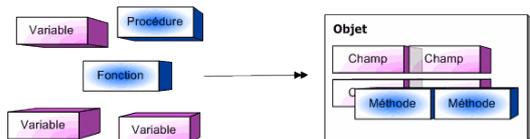
L'**encapsulation** introduit donc une nouvelle manière de gérer des données. Il ne s'agit plus de déclarer des données générales puis un ensemble de procédures et fonctions destinées à les gérer de manière séparée, mais bien de réunir le tout sous le couvert d'une seule et même entité.

Si l'**encapsulation** est déjà une réalité dans les langages procéduraux (comme le Pascal non objet par exemple) au travers des unités et autres bibliothèques, il prend une toute nouvelle dimension avec l'**objet**.

En effet, sous ce nouveau concept se cache également un autre élément à prendre en compte : pouvoir masquer aux yeux d'un programmeur extérieur tous les rouages d'un objet et donc l'ensemble des procédures et fonctions destinées à la gestion *interne* de l'objet, auxquelles le programmeur final n'aura pas à avoir accès. L'encapsulation permet donc de masquer un certain nombre de champs et méthodes tout en laissant visibles d'autres champs et méthodes.

Nous verrons ceci un peu plus loin.

Pour conclure, l'encapsulation permet de garder une cohérence dans la gestion de l'objet, tout en assurant l'intégrité des données qui ne pourront être accédées qu'au travers des méthodes visibles.



2.3.2. Héritage

Si l'encapsulation pouvait se faire manuellement (grâce à la définition d'une unité par exemple), il en va tout autrement de l'**héritage**. Cette notion est celle qui s'explique le mieux au travers d'un exemple. Considérons un objet *Bâtiment*. Cet objet est pour le moins générique, et sa définition reste assez vague. On peut toutefois lui associer divers champs, dont par exemple :

- les murs ;
- le toit ;
- une porte ;
- l'adresse ;
- la superficie.

On peut supposer que cet objet *Bâtiment* dispose d'un ensemble de méthodes destinées à sa gestion. On pourrait ainsi définir entre autres des méthodes pour :

- ouvrir le Bâtiment ;
- fermer le Bâtiment ;
- agrandir le Bâtiment.

Grâce au concept d'**héritage**, cet objet *Bâtiment* va pouvoir donner naissance à un ou des *descendants*. Ces descendants vont tous bénéficier des caractéristiques propres de leur *ancêtre*, à savoir ses champs et méthodes. Cependant, les descendants conservent la possibilité de posséder leurs propres champs et méthodes. Tout comme un enfant hérite des caractéristiques de ses parents et développe les siennes, un objet peut hériter des caractéristiques de son ancêtre, mais aussi en **développer de nouvelles**, ou bien encore se **spécialiser**.

Ainsi, si l'on poursuit notre exemple, nous allons pouvoir créer un objet *Maison*. Ce nouvel objet est toujours considéré comme un *Bâtiment*, il possède donc toujours des murs, un toit, une porte, les champs *Adresse* ou *Superficie* et les méthodes destinées par exemple à *Ouvrir le Bâtiment*.

Toutefois, si notre nouvel objet est toujours un *Bâtiment*, il n'en reste pas moins qu'il s'agit d'une *Maison*. On peut donc lui adjoindre d'autres champs et méthodes, et par exemple :

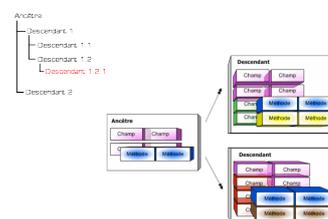
- nombre de fenêtres ;
- nombre d'étages ;
- nombre de pièces ;
- possède ou non un jardin ;
- possède une cave.

Notre *Bâtiment* a ici bien évolué. Il s'est **spécialisé**. Avec notre *Maison*, nous sommes allés plus avant dans les détails, et elle est à même de nous offrir des services plus évolués. Nous avons complété ce qui n'était qu'un squelette.

Ce processus d'héritage peut bien sûr être répété. Autrement dit, il est tout à fait possible de déclarer à présent un descendant de *Maison*, développant sa spécialisation : un *Chalet* ou encore une *Villa*. Mais de la même manière, il n'y a pas de restrictions théoriques concernant le nombre de descendants pour un objet. Ainsi, pourquoi ne pas déclarer des objets *Immeuble* ou encore *Usine* dont l'ancêtre commun serait toujours *Bâtiment*.

Ce concept d'**héritage** ouvre donc la porte à un nouveau genre de programmation.

On notera qu'une fois qu'un champ ou une méthode sont définis, il ou elle le reste pour tous les descendants, quel que soit leur degré d'éloignement.



2.3.3. Polymorphisme

Le terme **polymorphisme** est certainement celui que l'on appréhende le plus. Mais il ne faut pas s'arrêter à cela. Afin de mieux le cerner, il suffit d'analyser la structure du mot : *poly* comme plusieurs et *morphisme* comme *forme*. Le **polymorphisme** traite de la capacité de l'objet à posséder *plusieurs formes*.

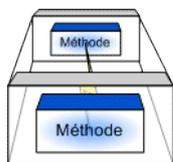
Cette capacité dérive directement du principe d'héritage vu précédemment. En effet, comme on le sait déjà, un objet va hériter des champs et méthodes de ses ancêtres. Mais un objet garde toujours la capacité de pouvoir **redéfinir une méthode** afin de la réécrire, ou de la **compléter**.

On voit donc apparaître ici ce concept de **polymorphisme** : choisir en fonction des besoins quelle méthode ancêtre appeler, et ce au cours même de l'exécution. Le comportement de l'objet devient donc modifiable à volonté.

Le **polymorphisme**, en d'autres termes, est donc la capacité du système à choisir dynamiquement la méthode qui correspond au type réel de l'objet en cours. Ainsi, si l'on considère un objet *Véhicule* et ses descendants *Bateau*, *Avion*, *Voiture* possédant tous une méthode *Avancer*, le système appellera la fonction *Avancer* spécifique suivant que le véhicule est un *Bateau*, un *Avion* ou bien une *Voiture*.

Attention !

Le concept de **polymorphisme** ne doit pas être confondu avec celui d'*héritage multiple*. En effet, l'héritage multiple - non supporté par le Pascal standard - permet à un objet d'hériter des membres (champs et méthodes) de plusieurs objets à la fois, alors que le **polymorphisme** réside dans la capacité d'un objet à modifier son comportement propre et celui de ses descendants au cours de l'exécution.



3. Différents types de méthodes

Parmi les différentes méthodes d'un objet se distinguent deux types de méthodes bien particulières et remplissant un rôle précis dans sa gestion : les **constructeurs** et les **destructeurs**.

3.1. Constructeurs et destructeurs

3.1.1. Constructeurs

Comme leur nom l'indique, les **constructeurs** servent à **construire l'objet en mémoire**. Un **constructeur** va donc se charger de mettre en place les données, d'associer les méthodes avec les champs et de créer le *diagramme d'héritage* de l'objet, autrement dit de mettre en place toutes les liaisons entre les ancêtres et les descendants.

Il faut savoir que s'il peut exister en mémoire plusieurs instances d'un même type objet, autrement dit plusieurs variables du même type, seule **une copie des méthodes** est conservée en mémoire, de sorte que chaque instance se réfère à la même zone mémoire en ce qui concerne les méthodes. Bien entendu, les champs sont distincts d'un

objet à un autre. De fait, seules les données diffèrent d'une instance à une autre, la "machinerie" reste la même, ce qui permet de ne pas occuper inutilement la mémoire.

Certaines remarques sont à prendre en considération concernant les constructeurs.

- Un objet peut **ne pas avoir de constructeur explicite**. Dans ce cas, c'est le compilateur qui se charge de créer de manière **statique** les liens entre champs et méthodes.
- Un objet peut avoir **plusieurs constructeurs** : c'est l'utilisateur qui décidera du constructeur à appeler. La présence de constructeurs multiples peut sembler saugrenue de prime abord, leur rôle étant identique. Cependant, comme pour toute méthode, un constructeur peut être *surchargé*, et donc effectuer diverses actions en plus de la construction même de l'objet. On utilise ainsi généralement les constructeurs pour **initialiser les champs** de l'objet. À différentes initialisations peuvent donc correspondre différents constructeurs.
- S'il n'est pas nécessaire de fournir un constructeur pour un objet statique, il devient **obligatoire** en cas de **gestion dynamique**, car le *diagramme d'héritage* ne peut être construit de manière correcte que lors de l'exécution, et non lors de la compilation.

3.1.2. Destructeurs

Le **destructeur** est le pendant du constructeur : il se charge de **détruire l'instance de l'objet**. La mémoire allouée pour le *diagramme d'héritage* est libérée. Certains compilateurs peuvent également se servir des destructeurs pour éliminer de la mémoire le code correspondant aux méthodes d'un type d'objet si plus aucune instance de cet objet ne réside en mémoire.

Là encore, différentes remarques doivent être gardées à l'esprit.

- Tout comme pour les constructeurs, un objet peut **ne pas avoir de destructeur**. Une fois encore, c'est le compilateur qui se chargera de la destruction **statique** de l'objet.
- Un objet peut posséder **plusieurs destructeurs**. Leur rôle commun reste identique, mais peut s'y ajouter la destruction de certaines variables internes pouvant différer d'un destructeur à l'autre. La plupart du temps, à un constructeur distinct est associé un destructeur distinct.
- En cas d'utilisation **dynamique**, un **destructeur s'impose** pour détruire le diagramme créé par le constructeur.

3.2. Pointeur interne

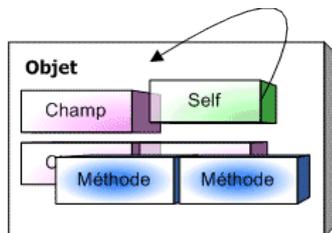
Très souvent, les objets sont utilisés de manière **dynamique**, et ne sont donc créés que lors de l'exécution. Si les méthodes sont toujours communes aux instances d'un même type objet, il n'en est pas de même pour les données.

Il peut donc se révéler indispensable pour un objet de pouvoir se référencer lui-même. Pour cela, toute instance

dispose d'un **pointeur interne** vers elle-même.

Ce pointeur peut prendre différentes appellations. En **Pascal**, il s'agit du pointeur **Self**.

D'autres langages pourront le nommer *this*, comme le C++...



3.3. Méthodes virtuelles et méthodes dynamiques

3.3.1. Méthodes virtuelles

3.3.1.1. Principe

Une méthode dite **virtuelle** n'a rien de fictif ! Il s'agit d'une méthode dont la résolution des liens est effectuée dynamiquement. Voyons ce que cela signifie.

Comme nous le savons déjà, toute méthode est susceptible d'être surchargée dans un descendant, de manière à être écrasée ou complétée. Par conséquent, toute méthode surchargée donne lieu à création d'une nouvelle section de code, et donc à une nouvelle adresse en mémoire.

De plus, tout objet possède un lien vers la table des méthodes de ses ancêtres : le diagramme d'héritage. De fait, tout type objet est directement lié à ses types ancêtres. Autrement dit, si nous reprenons l'exemple du début, l'objet *Maison* peut être assimilé à un *Bâtiment*.

Considérons à présent la méthode *Ouvrir* d'un *Bâtiment*. Celle-ci consiste à ouvrir la porte principale.

À présent, surchargeons cette méthode pour l'objet *Maison*, de sorte que la méthode *Ouvrir* non seulement ouvre la porte principale, mais également les volets de notre *Maison*.

Déclarons maintenant une instance **statique** de *Bâtiment*, et appelons cette méthode *Ouvrir*. Lors de la création de l'exécutable, le compilateur va vérifier le type d'instance créé. Le compilateur lie alors notre appel à celui de *Bâtiment.Ouvrir* (la méthode *Ouvrir* de l'objet *Bâtiment*), en toute logique. Il ne se pose aucun problème.

Considérons à présent un autre exemple : déclarons une variable **dynamique** destinée, en principe, à recevoir un objet *Bâtiment*. Comme nous l'avons vu juste avant, l'objet *Maison* est **compatible** avec l'objet *Bâtiment*. Comme nous travaillons en dynamique, nous nous servons de **pointeurs**. De fait, je peux très bien décider, avec cette variable pointant vers un objet *Bâtiment*, de déclarer une instance de type *Maison* : le compilateur ne montrera aucune réticence.

Si nous résumons, nous avons donc une variable de type officiel *pointeur* vers *Bâtiment* et contenant en réalité une *Maison*.

Appelons alors notre méthode *Ouvrir*. Comme nous avons une *Maison*, il faut que l'on ouvre les volets. Or, si nous exécutons notre programme, les volets resteront clos. Que

s'est-il passé ?

Lors de la création du programme, le compilateur s'est arrêté sur notre appel à *Ouvrir*. Ayant déclaré un objet *Bâtiment*, le compilateur **ignore tout du comportement du programme lors de son exécution**, et par conséquent ignore que la variable de type *pointeur* vers *Bâtiment* contiendra à l'exécution un objet *Maison*. De fait, il effectue une liaison vers *Bâtiment.Ouvrir* alors que nous utilisons une *Maison* !

La solution, vous l'aurez compris, réside dans l'utilisation des **méthodes virtuelles**. Grâce à celles-ci, la résolution des liens est effectuée **dynamiquement**, autrement dit **lors de l'exécution**. Ainsi, si nous déclarons notre méthode *Ouvrir* comme **virtuelle**, lors de la création du programme, le compilateur n'effectuera aucune liaison statique avant notre appel. Ce n'est que lors de l'exécution, au moment de l'appel, que la liaison va s'effectuer. Ainsi, au moment où l'on désirera appeler *Ouvrir*, notre programme va interroger son **pointeur interne** pour déterminer son type. Bien évidemment, cette fois-ci, il va détecter une instance de *Maison*, et l'appel se fera donc en direction de *Maison.Ouvrir*. Les volets s'ouvrent...

Vous aurez noté toute l'importance des méthodes virtuelles. D'une manière générale, **sitôt qu'une méthode est susceptible d'être surchargée, il faut la déclarer comme virtuelle**.

Attention !

Les **constructeurs** des **objets** ne seront **jamais déclarés comme virtuels**, car c'est toujours le bon constructeur qui est appelé. Le caractère virtuel est donc inutile et sera même signalé comme une erreur par le compilateur.

Par contre, les **destructeurs** seront **toujours déclarés comme virtuels** car souvent surchargés.

Il n'en est pas de même pour les **classes** qui elles peuvent s'appuyer sur le principe de *constructeur virtuel*. C'est notamment le cas de **Delphi** avec les références de classes à propos desquelles la documentation donne plus de précisions.

Vous pouvez aussi consulter les tutoriels suivants :

* Cours sur la POO de Frédéric Beaulieu : [Lien 230](#) ;

* Cours sur les métaclasse de Laurent Dardenne : [Lien 231](#).

3.3.1.2. Constructeurs et Table des Méthodes Virtuelles

Afin de pouvoir appeler la méthode appropriée au moment souhaité, un objet doit s'appuyer sur une liste de ses méthodes virtuelles : la **VMT** ou **Virtual Methods Table**, la Table des Méthodes Virtuelles. Cette table est mise en place par les **constructeurs** d'un objet.

Tout objet possède sa propre VMT, conservant toujours un lien avec la VMT de son ancêtre.

Lorsqu'un appel à une méthode virtuelle est effectué, l'objet recherche dans sa VMT s'il trouve la méthode recherchée. Si c'est le cas, alors il utilise l'adresse enregistrée et exécute la méthode. Sinon, il parcourt la VMT de son ancêtre direct et ainsi de suite jusqu'à l'ancêtre le plus éloigné dans la hiérarchie.

De même, lorsque qu'une méthode surchargée fait appel à la méthode ancêtre, alors une recherche est effectuée en partant cette fois-ci de la VMT du premier ancêtre.

La VMT est détruite par un **destructeur** lorsque celle-ci n'a plus lieu d'exister.

Si jamais on utilise une méthode **virtuelle** sans avoir appelé au préalable un **constructeur**, le caractère virtuel ne sera pas pris en compte et les résultats seront **imprévisibles**.

3.3.2. Méthodes dynamiques

Après les méthodes virtuelles, on se demande ce que l'on a pu inventer de pire ! Rassurez-vous, rien du tout. Les méthodes **dynamiques** ne sont en fait que des méthodes virtuelles.

Leur particularité réside dans le fait qu'elles sont **indexées**. Autrement dit, chaque méthode **dynamique** possède un numéro unique pour l'identifier.

Il convient de les comparer aux méthodes virtuelles :

- **Avantage** : les méthodes dynamiques consomment moins de mémoire ;
- **Inconvénient** : la gestion interne des méthodes dynamiques est plus complexe, et donc plus lente.

Par conséquent, **on préférera toujours les méthodes virtuelles, sauf si de nombreuses méthodes virtuelles doivent être déclarées, auquel cas on se reportera aux méthodes dynamiques**.

3.4. Méthodes abstraites

Une méthode **abstraite** est une méthode qu'il est **nécessaire de surcharger**. Elle ne possède donc **pas d'implémentation**. Ainsi, si on tente d'appeler une méthode **abstraite**, alors une erreur est déclenchée.

Bien entendu, il convient lors de la surcharge d'une telle méthode de ne pas faire appel à la méthode de l'ancêtre...

Les méthodes **abstraites** sont généralement utilisées lorsque l'on bâtit un squelette d'objet devant donner lieu à de multiples descendants devant tous posséder un comportement analogue. On pourra prendre notamment l'exemple de l'objet **TStream** et de tous ses descendants.

4. Visibilité

De par le principe de l'encapsulation, afin de pouvoir garantir la protection des données, il convient de pouvoir masquer certaines données et méthodes internes les gérant, et de pouvoir laisser visibles certaines autres devant servir à la gestion publique de l'objet. C'est le principe de la **visibilité**.

4.1. Champs et méthodes publics

Comme leur nom l'indique, les champs et méthodes dits **publics** sont accessibles depuis tous les descendants et dans tous les modules : programme, unité...

On peut considérer que les éléments **publics** n'ont pas de restriction particulière.

Les méthodes publiques sont communément appelées **accesseurs** : elles permettent d'accéder aux champs d'ordre *privé*.

Il existe des accesseurs en *lecture*, destinés à récupérer la valeur d'un champ, et des accesseurs en *écriture* destinés pour leur part à la modification d'un champ.

Il n'est pas nécessaire d'avoir un accesseur par champ privé, car ceux-ci peuvent n'être utilisés qu'à des fins internes.

Très souvent, les accesseurs en *lecture* verront leur nom commencer par *Get* quand leurs homologues en *écriture* verront le leur commencer par *Set* ou *Put*.

Les **constructeurs** et les **destructeurs** éventuels d'un objet devront bénéficier de la visibilité **publique**, sans quoi un programme externe ne pourrait pas les appeler !

Attention !

Un champ ne devra être public **que si sa modification n'entraîne pas de changement dans le comportement de l'objet**. Dans le cas contraire, il faut passer par une méthode. Modifier un champ "manuellement" et ensuite appeler une méthode pour informer de cette modification est une **violation du principe d'encapsulation**.

4.2. Champs et méthodes privés

La visibilité **privée** restreint la portée d'un champ ou d'une méthode au **module où il ou elle est déclaré(e)**. Ainsi, si un objet est déclaré dans une unité avec un champ privé, alors ce champ ne pourra être accédé qu'à l'intérieur même de l'unité.

Cette visibilité est à bien considérer. En effet, si un descendant doit pouvoir accéder à un champ ou une méthode privé(e), alors ce descendant doit nécessairement être déclaré dans le **même module que son ancêtre**.

Généralement, les **accesseurs**, autrement dit les méthodes destinées à modifier les champs, sont déclarés comme **privés**.

4.3. Champs et méthodes protégés

La visibilité **protégé** correspond à la visibilité **privé** excepté que tout champ ou méthode protégé(e) est accessible dans tous les descendants, quel que soit le module où ils se situent.

Cette visibilité est souvent à préférer à la visibilité privée, cependant elle n'est pas supportée par tous les compilateurs. Par exemple, **Turbo Pascal ne la reconnaît pas**.

5. Le Pascal Objet

Tous les éléments de la Programmation Orientée Objet énoncés jusqu'ici sont bien entendu supportés par le Pascal Objet. Nous allons voir à présent comment les implémenter.

5.1. Déclaration d'un objet

5.1.1. Déclaration simple

Avant de pouvoir utiliser la POO, il convient de savoir

déclarer un objet. Pour ce faire, on a recours au mot réservé **object**. Un objet étant une *structure de données*, il sera donc toujours déclaré à l'intérieur d'un bloc **type**.

L'objet le plus basique que l'on puisse déclarer est tout simplement l'objet vide, qui n'effectue rien et ne contient rien. On le déclare comme ceci :

```
type
  TObjetVide = object
end;
```

On remarquera que la déclaration d'un objet se terminera **toujours** par un **end;**.

De plus, on adopte très souvent en Pascal une convention de notation pour les objets : leur nom commence toujours par **T**, comme *type*. Nous respecterons toujours cette convention dans ce tutoriel, et nous parlerons donc des objets *TMaison* ou encore *TChose*.

Attention !

Sur les compilateurs de nouvelle génération, le mot réservé **object** sera remplacé par le mot réservé **class**.

```
type
  TObjetVide = class
end;
```

Dans la suite de ce tutoriel, nous adopterons l'utilisation de mot réservé **object**. Celui-ci sera à remplacer en fonction du compilateur utilisé (**Delphi** par exemple).

5.1.2. Déclarations imbriquées

Il est parfois nécessaire de déclarer des objets qui s'utilisent mutuellement. On peut ainsi prendre l'exemple de deux objets, *TParent* et *TEnfant*, le parent ayant la nécessité de connaître la liste de ses enfants, et l'enfant la nécessité de connaître son parent.

Intervient alors le problème de la déclaration imbriquée de deux objets. Si l'on tente de les déclarer comme ceci :

```
type
  TParent = object
    Enfant: TEnfant;
end;

  TEnfant = object
    Parent: TParent;
end;
```

Le compilateur va déclencher une **erreur** indiquant qu'il

ne connaît pas encore *TEnfant* lorsqu'il tente d'analyser la structure de *TParent*.

La solution au problème passe **nécessairement par l'instanciation dynamique** des deux objets. On va donc déclarer deux pointeurs vers les deux types considérés et seulement ensuite, on déclarera les objets eux-mêmes, ceci dans **le même bloc type** :

```
type
  { Déclaration des pointeurs }
  PParent = ^TParent;
  PEnfant = ^TEnfant;

  { Déclaration des objets utilisant les pointeurs }
  TParent = object
    Enfant: PEnfant;
end;

  TEnfant = object
    Parent: PParent;
end;
```

Grâce à cette méthode, plus aucune erreur n'est déclenchée, car lorsque le compilateur va analyser *TParent*, il aura déjà eu connaissance de l'existence de *PEnfant*. Ceci n'est bien évidemment possible que parce que le compilateur accepte la déclaration **prématurée** de pointeurs avant le type vers lequel ils pointent.

Sous **Delphi**, l'instanciation étant automatiquement **dynamique**, ce problème ne se pose pas, et pour résoudre le problème, on se contente d'**annoncer la classe** avec une **déclaration partielle** :

```
type
  { Déclarations partielles }
  TParent = class;
  TEnfant = class;

  { Déclaration des objets utilisant les classes }
  TParent = class
    Enfant: TEnfant;
end;

  TEnfant = class
    Parent: TParent;
end;
```

Retrouvez la suite de l'article d'Eric Sigoillot en ligne : [Lien 232](#)

Bien débuter en Perl

Vous désirez apprendre le langage Perl ? 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. Je vous propose également de nombreux liens pour vous permettre de vous exercer et d'aller plus loin !

1. Qu'est-ce que Perl ?

Perl est un langage de programmation conçu par Larry Wall dont la première version date de 1987. Ce langage de programmation a été créé dans l'optique d'être un intermédiaire entre le langage C et les langages de scripts comme bash. Au premier, il emprunte la syntaxe et au second la simplicité.

Perl est idéal pour :

- l'administration système ;
- la manipulation de texte (mail, logs...) ;
- la création d'interfaces graphiques simples ;
- faire l'interface avec un système de gestion de base de données (MySQL, Oracle, PostgreSQL...) ;
- l'automatisation de tâches répétitives ;
- la programmation Web, avec des modules CGI ;
- et bien d'autres choses encore !

Cerise sur le gâteau, Perl est multiplateforme ! Que vous travailliez sur un Unixöide (Linux, BSD, Solaris, Mac...) ou un Windows, votre script Perl tournera sans aucune modification sur tous ces systèmes.

Au niveau technique, Perl présente les avantages suivants :

- gestion automatique de la mémoire (système de Garbage Collector) ;
- manipulation aisée de données hétérogènes : tableau, chaîne de caractères, table de hachage, référence, et le plus important, tout cela combiné !
- gestion native des expressions régulières ou regex ;
- plus de 20000 modules disponibles sur le site du CPAN.

Pour finir cette présentation, je ne peux m'empêcher de vous donner le slogan de Perl, qui témoigne bien de son état d'esprit : There Is More Than One Way To Do It (il y a plus d'une façon de le faire).

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

2.1. L'éditeur de texte

Pour débuter en Perl, la première chose est de disposer d'un bon éditeur de texte, qui permettra notamment de mettre en couleur vos codes.

Sous Windows, je conseille Notepad++ ([Lien 233](#)), qui est un logiciel libre souvent mis à jour et qui offre beaucoup de possibilités tout en restant simple à utiliser.

Sous Linux, si vous n'aimez pas les éditeurs comme vim

ou emacs qui se basent surtout sur la ligne de commande, je vous conseille l'excellent SciTe ([Lien 234](#)). On notera qu'il est également disponible sous Windows.

Sous Mac, un excellent éditeur en français est Smultron ([Lien 235](#)), aujourd'hui remplacé par l'éditeur Fraise ([Lien 236](#)).

Notez bien que les logiciels proposés ne sont pas les seuls, loin de là. Si vous avez déjà un éditeur favori qui gère la coloration syntaxique, utilisez-le !

2.2. Perl

Une fois votre éditeur configuré, attaquons-nous au principal : Perl lui-même ! Si vous êtes sous Linux ou Mac, réjouissez-vous : il y a 95 % de chances pour que Perl soit déjà présent sur votre système ! Si vous n'en êtes pas sûr, lancez une console et tapez « perl -v ». Si Perl est installé, la commande affichera la version présente sur le système.

Si Perl n'est pas installé, deux écoles : le recompiler vous-même (les sources sont disponibles sur perl.com : [Lien 237](#)) ou suivre les instructions pour télécharger et installer ActivePerl, disponible sur le site d'ActiveState ([Lien 238](#)) dans la section ActivePerl ([Lien 239](#)). Sous Windows, choisissez le fichier MSI, sous Mac le DMG et sous Linux le .DEB ou le .RPM.

2.3. Test de fonctionnement

Ouvrez votre éditeur, choisissez le langage Perl dans les choix de coloration syntaxique et tapez le code suivant :

```
#!/usr/bin/perl -w
use strict;
use warnings;

print "Hello World !\n";
```

Enregistrez ce code sous le nom de test.pl (les fichiers Perl ont une extension .pl ou .pm pour les modules). Rendez ce programme exécutable (chmod +x test.pl).

Ensuite, lancez une console ou invite de commande, placez-vous dans le répertoire où vous avez enregistré le fichier (à l'aide de la commande cd) et tapez :

```
perl test.pl
```

Normalement, le texte *Hello World !* devrait s'afficher à l'écran. Félicitations ! Vous venez de faire votre premier programme avec Perl !

3. Quelles sont les ressources disponibles pour apprendre le langage Perl ?

3.1. Les cours en ligne

Maintenant que vous disposez de tous les outils nécessaires pour faire du Perl, il est temps d'apprendre le langage. Avant de vous acheter un livre sur le sujet, vous préférerez peut-être suivre un cours d'introduction à Perl en ligne et gratuit. Là encore, deux écoles :

- si vous voulez rapidement être opérationnel sans aborder certaines subtilités, je vous conseille *Formez-vous rapidement à Perl*, partie 1 ([Lien 240](#)), 2 ([Lien 241](#)) et 3 ([Lien 242](#)), de David Bourgeois ;
- si vous souhaitez une introduction plus complète au langage Perl, Sylvain Lhullier a rédigé une excellente Introduction à la programmation en Perl ([Lien 243](#)), que vous pouvez suivre même si vous n'avez aucune connaissance en programmation.

Une fois ces cours assimilés, tous nos cours sont disponibles sur ce site : [Lien 244](#).

3.2. Les livres

Ça y est, vous avez appris le langage Perl et bien évidemment vous en êtes tombé amoureux ! Vous avez décidé d'acheter un livre, mais vous ne savez pas lequel ! Il se trouve qu'Oreilly propose deux excellents livres sur le sujet, très complémentaires :

- Introduction à Perl de Schwartz Randal, Phoenix Tom et Foy Briab. C'est le livre idéal si vous êtes débutant : [Lien 245](#);
- Programmation en Perl, de Larry Wall, Tom Christiansen et Jon Orwant. Ce livre (surnommé Camel Book) est l'ouvrage de référence sur Perl, à posséder absolument : [Lien 246](#).

D'autres livres sur Perl sont référencés sur ce site : [Lien 247](#).

3.3. Pour aller plus loin

Je vous encourage également à regarder nos cours ([Lien 248](#)) et notre FAQ ([Lien 249](#)).

4. J'ai un problème, aidez-moi !

Pas de panique, le club des développeurs est là pour vous aider.

Tout d'abord, pensez bien à regarder dans la documentation officielle pour vérifier si votre problème n'est pas déjà abordé. Tapez perldoc -h dans une console et laissez-vous guider !

Vous n'arrivez toujours pas à régler votre problème ? C'est le moment d'utiliser notre FAQ de plus de 150 questions ([Lien 249](#)), avec moteur de recherche intégré !

Le problème persiste ? Visitez notre forum et posez votre question ([Lien 250](#)) !

5. Quelques liens utiles

Pour finir, voici une liste des liens les plus utiles :

- Le site officiel de Perl. Vous y trouverez la documentation officielle, l'interpréteur Perl et bien d'autres choses encore : [Lien 237](#) ;
- Le site de la communauté internationale Perl : [Lien 251](#);
- Le site des traducteurs de la documentation Perl en français : [Lien 252](#) ;
- Le site où vous trouverez tous vos modules Perl : [Lien 253](#).

6. Conclusion

Voilà, ce guide touche à sa fin ! J'espère qu'il vous aura aidé à faire vos premiers pas en Perl et qu'il aura suscité votre intérêt pour ce merveilleux langage !

Retrouvez l'article de François Lieuze en ligne : [Lien 254](#)

Introduction au développement en équipe pour la 3D

Cet article explique comment aborder un développement en équipe pour la 3D. Il vous expliquera les mesures de base à mettre en place puis illustrera concrètement comment les mettre en place au travers d'un exemple concret : le projet AITC.

1. Objectifs de l'article

Cet article a pour but d'illustrer concrètement quelques méthodes basiques pour le développement de 3D interactive en équipe. Ces méthodes s'appliquent au jeu vidéo mais aussi à la réalité virtuelle, réalité augmentée ou tout autre domaine où il y a de la 3D interactive. On va aborder ensemble la vaste question :

« Comment organiser efficacement un développement en équipe ? »

En effet, on travaille toujours en équipe quand on recherche une certaine qualité. Dans le cas d'une équipe d'amateurs de jeu vidéo, le noyau dur sera constitué d'un graphiste 2D/3D, d'un programmeur et d'un level/game designer. Parmi eux, le plus expérimenté (ou le moins con) sera désigné comme chef de projet et aura comme tâche d'organiser le développement et d'affecter les tâches.

Sans ce minimum de compétences, il sera difficile de réaliser quelque chose de sympa. Par ailleurs, on remarque que ces compétences ne sont pas cumulées par la même personne.

Vous serez donc bien obligé de bosser en équipe... Alors autant savoir ce qui vous attend.

Le plus important est de s'attacher à ce que les différents intervenants du projet travaillent ensemble de manière coordonnée et sans gaspiller trop d'énergie. Il faudra éviter que Gérard écrase par inadvertance le boulot que François vient de faire dans le décor du niveau, ou que Malik perde son temps à coder un comportement pour la caméra alors que Simon a déjà développé quelque chose d'intéressant.

Ce sont avant tout des problèmes d'organisation et de communication à la charge du chef de projet. Mais il existe des outils et des méthodes qui peuvent grandement simplifier sa tâche et celle de toute l'équipe.

Dans la littérature très abondante qui existe sur le sujet de la gestion de projet, il existe assez peu d'exemples concrets qui concernent la 3D. De l'autre côté les articles techniques ne discutent pas de la gestion de projet.

Nous avons donc fait le choix en rédigeant cet article d'avoir une approche mixte en illustrant très concrètement chacune de nos propositions en les appliquant à un exemple de projet nommé « Alone In The Church » (AITC). Pour tirer le maximum de cet article, le lecteur devra être déjà initié.

Espérons que tout cela puisse vous servir de base pour votre premier développement amateur en équipe (entre 3 et 10 personnes maximum).

Les logiciels utilisés sont 3dsMax 9, Virtools 3.5 et SVN. Les principes exposés ici sont suffisamment généraux pour s'adapter à d'autres moteurs.

L'exposé sera organisé en grands chapitres. Dans un premier temps nous allons analyser ce qu'est un développement en équipe pour déterminer ce qu'il faudrait mettre en place pour assurer sans tracas. Nous verrons ensuite comment cela se traduit chacune des grandes étapes de développement.

2. Approche générale : le développement en équipe

2.1. Qu'est-ce qu'un développement en équipe ?

Précisons notre sujet : un développement est un processus de production numérique. Pour produire une application 3D, il faut suivre les étapes principales suivantes :

Les principales étapes de développement :

1. **la conception** pour fixer les objectifs et les méthodes employées lors du projet ;
2. **l'implémentation** pour réaliser les composants du projet ;
3. **l'intégration** pour associer les différents composants du projet ;
4. **les tests** pour vérifier que le comportement des composants assemblés est bien celui attendu.

Même si ce processus est présenté ici comme une suite fixe d'étapes, dans les faits ce n'est pas un enchaînement statique, parfaitement prédéfini. En effet, dans tout processus de création, y compris numérique, des ajustements, des modifications, des corrections seront effectués. C'est obligatoire : votre personnage, votre décor, une interface ou même un bug vont connaître plusieurs versions. Dans certains cas, et si vous n'êtes pas chanceux et/ou expérimenté, il y aura même certains retours en arrière qui provoqueront des pertes de temps considérables.

Vous l'aurez compris, les étapes décrites plus haut ne sont là que pour rationaliser votre approche et gagner en efficacité. Vous pourrez ensuite investir l'énergie économisée sur les aspects que vous jugerez essentiels : la jouabilité, l'originalité des graphismes ou l'adéquation entre le gameplay et les situations décrites par le scénario.

OK, mais comment fait-on ?

Et bien, il faut faciliter le « work flow », c'est-à-dire le « flux de travail » littéralement. On doit considérer le travail réalisé comme un liquide ayant une source, une destination et un débit. En s'assurant que le travail de chacun des membres de l'équipe démarre au bon endroit et fini bien là où il doit finir (dans votre application et pas à la poubelle), vous gardez un développement continu sans (trop) de surprise.

Pour assurer cette fluidité, nous allons nous attacher à ce qu'une modification puisse être effectuée et répercutée en consommant le moins d'énergie possible. Cela sous-entend qu'on doit aussi pouvoir simplement annuler cette correction si elle n'est pas satisfaisante malgré vos efforts.

Compte tenu de la quantité et de la complexité des données manipulées, c'est loin d'être évident. Mais si vous inspirez des mesures qui ont cours dans le milieu professionnel et que vous les mettez correctement en place, c'est tout le processus de création qui sera facilité, du graphiste au codeur.

Voyons ensemble ces principales mesures en gardant à l'esprit que « Faire un jeu est tout sauf un jeu ! » comme dirait l'autre.

2.2. Comment gagner en efficacité ?

Les principales mesures pour s'assurer que les modifications sont simplement apportées au projet sont les suivantes :

- **partager et sécuriser les données ;**
Chaque membre de l'équipe doit avoir accès immédiatement à tous les documents du projet qui lui sont nécessaires, y compris la documentation.
- **gérer les versions des fichiers ;**
Pour chaque élément du projet, on doit connaître sa version actuelle et qui l'a modifié en dernier.
- **diviser pour régner ;**
On doit faire en sorte que chaque élément du projet ne traite que d'un domaine limité et soit réutilisable. Tout grand problème devra être divisé en sous-problèmes plus simples.
- **compiler quotidiennement**
On doit pouvoir compiler le projet tous les jours pour connaître son état d'avancement et pouvoir évaluer où l'on en est.

Ces mesures sont valables pour tous les éléments du projet : modélisation, texture, animation, documents de conception, programmes... Tout, et pas d'exception.

Abordons tout cela dans le détail.

3. L'efficacité du développement point par point

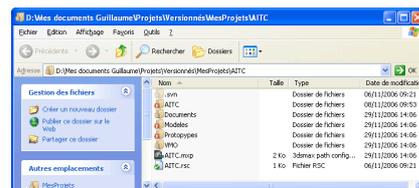
3.1. Partager et sécuriser les données

L'ensemble des informations doit être accessible par tous les développeurs. Cela veut dire qu'on doit savoir si l'information qu'on cherche existe et, si oui, où elle se trouve. On pourra ainsi accéder aux documents de conception, à la bibliothèque de matériaux, à la

documentation d'un composant ou tout autre élément du projet.

Concrètement, votre projet est un simple répertoire qui doit contenir au maximum dix sous-répertoires, eux-mêmes ne contenant que dix sous-répertoires et ainsi de suite. Tous les répertoires doivent avoir un nom explicite. L'arborescence précise dépend de la nature du projet mais on peut déjà dire qu'il prendra la forme suivante :

- MonProjetQuiDéchireTout
 - DocumentsDeConception
 - contient les documents de conception : Références, scénario, design et ainsi de suite.
 - Ressources Virtools du projet
 - contient tous les composants de votre projet 3D, les textures, les modèles 3D les comportements de caméra...
 - Modélisations
 - contient les fichiers de conception de vos modèles 3D de votre projet, c'est-à-dire les fichiers 3DSMax.
 - Graphisme2D
 - contient les fichiers conception de vos graphismes, c'est-à-dire les fichiers Photoshop ou autre.
 - Compilation
 - contient la dernière version compilée de votre projet, sous forme de fichiers VMO pour notre exemple.
 - Prototypes
 - contient les prototypes de vos composants.



Le répertoire du projet AITC

Dans la figure ci-dessus, le répertoire AITC contient les ressources du projet.

Ce répertoire doit être accessible à tous les membres de l'équipe via un serveur (site internet, ftp, répertoire sur un réseau local) et doit être sauvegardé automatiquement à la fin de chaque journée de travail. Ainsi vous serez assuré qu'en cas de problème seule une journée de travail sera perdue.

Ne sous-estimez pas ce point, car les disques durs vacillent bien plus souvent qu'on ne veut le croire. Qui plus est, plus vous aurez à bosser et plus votre matériel sera mis à rude épreuve, et plus il aura de chance de flancher.

Pour effectuer cette sauvegarde automatiquement, utilisez un logiciel comme Cobian Backup (qui est gratuit) et sauvegardez vers un autre ordinateur. Vos données doivent

toujours être en deux lieux différents. Ne stockez que les cinq dernières versions de votre arborescence en fonction de la place disponible.

Bien, vous avez enfin un répertoire distant qui contient la dernière version de votre projet : c'est le véritable dossier de votre projet. À partir de maintenant, vous ferez une copie sur votre poste de travail et travaillerez en local. Mais comment s'assurer que vous n'oublierez pas d'enregistrer les composants que vous modifierez ? C'est là qu'intervient le système de gestion de version.

3.2. Les systèmes de gestion de version

Le développement informatique est bien plus vieux que le jeu vidéo et un ensemble d'outils très performants a été mis au point pour vous faciliter la vie. Alors autant en profiter.

3.2.1. Qu'est-ce que la gestion de version ?

Comme son nom l'indique, c'est un procédé qui permet de gérer plusieurs versions d'un même fichier. En gros, chaque membre de l'équipe peut récupérer la dernière version des fichiers d'un projet, modifier le fichier de son choix et le livrer une fois le travail terminé.

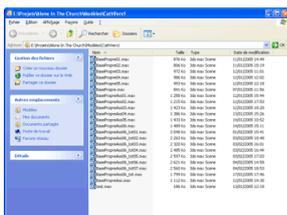
3.2.2. En quoi est-ce utile pour développer de la 3D ?

C'est une technique ancienne qui date des systèmes UNIX et de CVS. Même si le monde de la 3D a mis du temps à s'approprier ces outils, ils sont maintenant considérés comme communs, voire indispensables dans le cadre d'une production sérieuse. Par exemple, 3dsMax9 est livré avec un système de gestion de version « Vault Data Management ». Virtools quant à lui est compatible avec « AlienBrain ». Quant à nous, nous avons choisi d'utiliser SVN qui a le bon goût d'être gratuit, très performant, et compatible avec n'importe quel type de fichier (et pas seulement avec de la 3D).

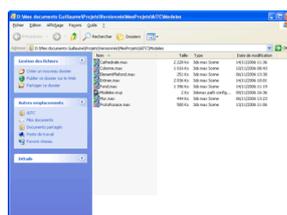
Il faut bien comprendre qu'un système de gestion de version est un outil incontournable pour une équipe qui souhaiterait obtenir des résultats de qualité. Car qui dit « qualité » dit aussi « organisation et outils ».

On va évoquer rapidement les avantages et la souplesse que ces outils apportent, et qui leur ont permis de s'imposer.

Un intérêt assez visible consiste dans la simplification de l'arborescence de travail du projet : on ne se traîne pas les vingt-cinq versions d'un fichier.

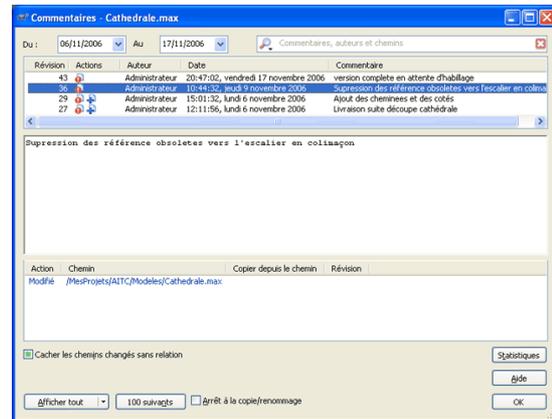


Sans gestion des versions



Avec gestion des versions

Même si on dispose d'un unique fichier, on peut à tout moment récupérer l'historique des modifications effectuées ainsi que n'importe quelle version antérieure.



Exemple d'historique d'un fichier .max

Dans certains cas, on peut aussi effectuer la fusion de modifications entre plusieurs versions d'un même fichier.

Par ailleurs, on peut bloquer les modifications sur un fichier pendant que l'on travaille dessus. De cette manière, vous pouvez empêcher un autre membre de l'équipe d'y accéder et de créer une version concurrente à la vôtre. Notez au passage que si deux membres de l'équipe travaillent sur le même fichier, le chef de projet n'a pas fait correctement son travail d'affectation des tâches.

Les fonctionnalités sont très nombreuses, et nous ne les citerons pas toutes. Mais pour finir soulignons le fait que le système SVN peut fonctionner à distance. Les données de l'équipe qui sont centralisées sur le serveur distant peuvent être récupérées via le net suivant différents protocoles, dont plusieurs sécurisés. La classe quoi.

3.2.3. Et si je veux m'y mettre ?

Comme dit plus haut, nous avons adopté pour cet article SVN (le successeur de CVS) et TortoiseSVN pour une intégration dans l'explorateur Windows.

Vous pouvez simplement l'installer chez vous en suivant les explications détaillées disponibles à l'adresse suivante : [Lien 255](#).

Attention, même si ces outils sont performants ils ne se substituent pas au travail de gestion de projet. On ne doit pas oublier d'autres mesures comme les conventions de nommage, ou l'affectation des tâches et des délais...

3.2.4. Application concrète : une infrastructure

Concrètement, vous devrez disposer de l'infrastructure suivante :

- un **serveur distant** pour :
 - stocker votre projet sous la forme d'un répertoire géré par SVN ;
 - effectuer les sauvegardes quotidiennes du projet vers une autre destination grâce à CobianBackup.
- des **postes de travail client** équipés :
 - d'un répertoire local de travail où sera placée la dernière version du projet par SVN ;
 - des logiciels de travail : 3dsMAX, Virtools, Photoshop...

Cette infrastructure peut sembler conséquente au premier abord. Toutefois, si on considère les objectifs de qualité à atteindre et le fait que vous travaillez en équipe, elle n'est pas si conséquente et reste accessible à des amateurs éclairés.

Enfin, si l'effectif de l'équipe devient important (supérieur à trois), une petite amélioration peut consister à héberger sur votre serveur le site internet de l'équipe. Réalisé sous la forme d'un site de news, il vous permettra d'assurer la communication du projet, en interne et en externe.

3.3. Diviser pour mieux régner

Quand un fichier va contenir une trop grande quantité de données, il faut le scinder en plusieurs fichiers de plus petite taille.

Le contenu de chaque sous-fichier sera beaucoup plus simple que le contenu du fichier original. Par conséquent, il sera beaucoup plus simple de s'y retrouver, pour intervenir et maintenir le fichier.

Par ailleurs, on permettra ainsi à plusieurs personnes de travailler en même temps sur ces données.

A priori, les critères pour décider si on divise un fichier sont :

- s'il devient difficile à manipuler ;
- si plusieurs personnes doivent intervenir régulièrement sur son contenu.

Par habitude, le moment de scinder un fichier est toujours repoussé. L'intérêt n'est pas évident au premier abord, mais une fois dans une période de stress vous râlerez abondamment sur ce p*** de fichier ingérable. N'attendez pas que ça se produise, anticipez si vous le pouvez.

L'autre aspect de cette mesure est de vous amener à produire des composants indépendants et donc **réutilisables**. C'est excessivement important, car cela vous permettra d'utiliser vos composants dans un autre projet. De cette manière vous gagnerez un temps précieux.

3.3.1. Application : le décor

Prenons, comme par hasard, l'exemple d'un décor représentant une cathédrale. On va scinder ce fichier car le décor devient trop grand. On est constamment obligé de cacher des parties pour s'y retrouver et on perd un temps précieux.

On décide donc de prendre certaines parties de ce décor pour les placer dans d'autres fichiers. L'entrée sera placée dans un fichier à part, la nef aussi ainsi que les ailes. Le fichier original existera toujours, mais il ne contiendra simplement que des liens vers les autres fichiers, rendant ainsi la maintenance bien plus simple.

Nous y reviendrons en détail dans la seconde partie de cet article.

3.4. La compilation quotidienne

On part du principe qu'on doit dépenser le minimum d'effort pour apporter une correction ou la supprimer. Il

faut avant tout éviter les réflexions du type « Modifier l'animation de marche du personnage ? P***tain... entre le .max et le moteur de jeu, je vais encore y passer 3 plombes. ».

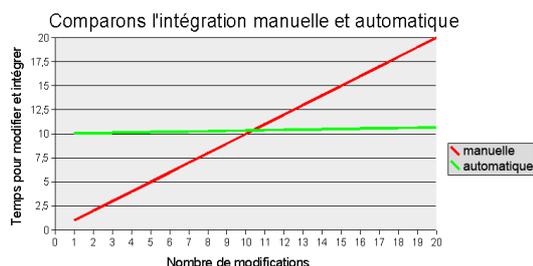
Dans une telle situation ce n'est pas tant la modification qui est fastidieuse mais de la répercuter. Il va falloir naviguer dans des dizaines de menus pour spécifier le nouveau fichier à charger à la place de l'ancien...

On doit absolument éviter cette situation qui consiste à avoir peur d'apporter une modification. Si c'est le cas, c'est le moment de tirer la sonnette d'alarme : vous êtes en train de refuser d'améliorer votre propre projet.

Votre solution : l'intégration automatique. Elle est plus difficile à mettre en place au début du projet, mais vous ne pourrez plus vous en passer par la suite.

Pour vous en convaincre, comparons ensemble le temps nécessaire pour intégrer une modification dans le cas où le système d'intégration est automatique ou manuel.

On suppose que réaliser une modification prend cinquante minutes et l'intégrer manuellement prend dix minutes soit un temps total d'une heure. On suppose aussi que l'intégration automatique demande dix heures d'efforts pour être mise en place, mais permet ensuite d'intégrer en seulement deux minutes. Si on réalise un graphique pour comparer les temps pour réaliser une modification et l'intégrer manuellement ou automatiquement, on obtient le résultat suivant :



On voit très clairement que dès qu'on fait plus de dix modifications, l'intégration manuelle prend toujours plus de temps que l'intégration automatique. Et plus vous ferez de modifications et plus il sera intéressant de mettre en place un système d'intégration automatique.

Effectivement cet exemple est un peu artificiel, mais réfléchissez : êtes-vous sûr que chaque membre de l'équipe fera moins de vingt modifications dans tout le projet ? Multipliez ça par le nombre de membres de l'équipe et il devient évident qu'un système d'intégration automatique est rentable très vite.

3.4.1. Application : le comportement de la caméra

Malik travail sur la gestion de la caméra. Depuis deux jours il travaille beaucoup : l'équipe est en retard. Heureusement pour lui, il vient de terminer son composant et part chercher un café bien mérité. À son retour devant son PC, il teste son composant et a une nouvelle idée. Il décide de la tester en codant une nouvelle version de son composant caméra. Il le fait sereinement car il sait que même si son nouveau composant n'est pas bon, il pourra

revenir à la version précédente en très peu de temps. Par contre, si son composant est bon il peut participer à augmenter la qualité du projet. En plus, Malik aura une bonne image de lui ; l'équipe sera rassurée de voir qu'elle peut produire plus que le strict nécessaire. Et tout cela va participer à détendre l'atmosphère et instiller une bonne ambiance : ce qui ne peut être que bon pour le projet.

Grâce à l'intégration automatique, substituer deux versions d'un composant est quasiment instantané.

En supprimant l'étape d'intégration manuelle on arrive plus vite à la phase de test. Ainsi on peut remarquer plus tôt les défauts et les corriger aussi plus rapidement. On favorise donc la qualité.

Par ailleurs, remarquez que le temps nécessaire à une intégration manuelle n'est plus gaspillé. Si la personne qui apportait la modification est compétente ça aurait été un véritable gâchis.

Maintenant vous pouvez réinvestir ce temps économisé ailleurs : babyfoot, machine à café, ou médire des chefs.

4. étude d'un cas

Maintenant que nous avons exposé les mesures à prendre, il est temps de passer à l'application. Pour rester efficaces et éviter de nous perdre dans une trop grande complexité nous allons nous cantonner à une partie du projet : le décor.

On souhaite réaliser un décor de niveau de jeu pour effectuer une démonstration technique de notre savoir-faire. Le projet est précisé dans un document de conception de base : le synopsis.

Synopsis du projet AITC

On souhaite réaliser une démonstration de notre savoir-faire en réalisant une visite virtuelle d'un « **château-cathédrale** »

Le décor devra sembler crédible, vieux et impressionnant. L'accent sera mis sur la qualité graphique et, dans cette optique, on devra utiliser tous les procédés disponibles (lightmaps, normal mapping et shaders).

En termes d'interactions, ce sera très simple : l'utilisateur pourra déplacer une caméra sans contraintes particulières. Elle devra simplement afficher les statistiques courantes pour évaluer les performances de l'application.

Vous noterez que ce synopsis n'est pas trop détaillé. L'équipe n'a pas de vision précise et sait que le projet évoluera en fonction des premiers résultats. Pour autant elle ne souhaite pas perdre le travail qu'elle aura effectué pour en arriver là.

Dans la partie suivante, nous allons suivre le travail des membres de l'équipe en charge du graphisme. On suppose que le reste de l'équipe va fournir les composants qui n'auront pas été produits dans cette partie.

4.1. Étude des graphismes : le décor

Nous allons voir comment chacune des quatre étapes de développement a été abordée par l'équipe des graphistes, et comment se traduisent concrètement les préconisations du chef de projet.

4.4.1. Conception : étude et design du décor

Sur la base de l'orientation définie par le synopsis, une banque d'images de références a été constituée pour réaliser le décor. Cette banque d'images est stockée dans un sous-répertoire « EtudeGraphiqueDecor » dans le répertoire « DocumentsDeConceptions ». Un fichier texte contenant les réflexions des graphistes sur le sujet est aussi créé.

Une première partie de cette banque d'images est constituée de photographies réalisées dans un lieu s'approchant correctement de l'esprit du projet, ici la cathédrale Saint-Pierre de Poitiers. Ces photos permettent d'approcher dans le même temps le décor dans sa globalité et les détails qui le composent.

Ces photos sont stockées dans un sous-répertoire « ReferencesPoitiersChatedrale » de manière à les dissocier des autres images de référence.

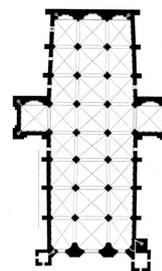


Références générales du projet AITC

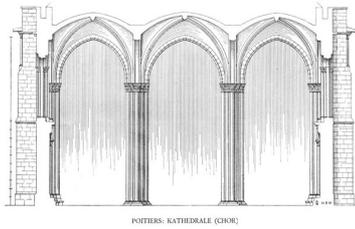
Cette approche générale étant satisfaisante, l'équipe complète sa vision du décor grâce à des plans architecturaux qu'elle stocke parmi les documents de conception dans le sous-répertoire « RéférencesPoitiersPlans ».

L'arborescence devient donc :

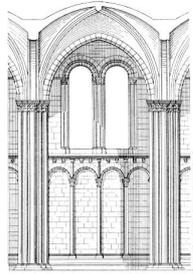
- « DocumentsDeConceptions »
 - « EtudeGraphiqueDecor »
 - « ReferencesPoitiersChatedrale »
 - « RéférencesPoitiersPlans »
 - Decor.doc



Références détaillées : plan au sol



Références détaillées : coupe transversale



Références détaillées : coupe longitudinale

À cette étape, l'équipe des graphistes considère que la conception est suffisamment avancée pour se lancer dans la modélisation du décor.

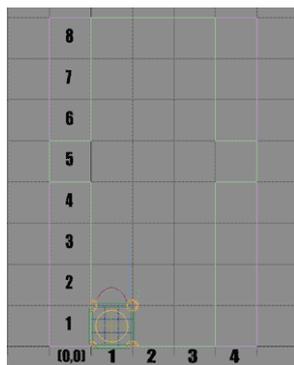
4.1.2. Réalisation : modélisation du décor

L'objectif est de trouver le bon compromis entre qualité visuelle et performances.

L'approche adoptée consiste donc à limiter le nombre de polygones présents dans la scène. Les formes générales vont être simplifiées, et des textures seront préférées à la géométrie pour réaliser les détails. Enfin, comme on considère que l'utilisateur ne pourra se déplacer que dans l'intérieur de la cathédrale, les faces qui ne seront jamais vues seront supprimées.

Toujours pour simplifier la modélisation et limiter le nombre de faces, l'équipe choisit d'appliquer la stratégie du « diviser pour régner ». En effet, une cathédrale est un cas assez évident de motifs répétés. L'équipe va donc utiliser des éléments de base qui seront ensuite multipliés pour créer le décor.

Enfin, une simplification générale du plan réel est effectuée pour obtenir le plan du décor modélisé. Sur ce plan, un repère de référence est mis en place sous la forme d'un damier pour placer et retrouver facilement un objet particulier.



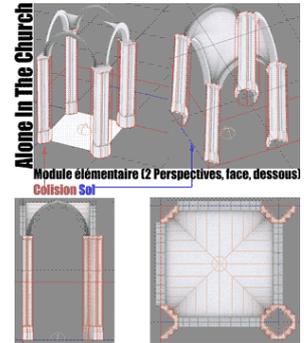
Plan du décor modélisé

Dans ce plan, un module de base est présent 3x8 fois à quelques détails près.

Ce module est constitué de deux composants de base :

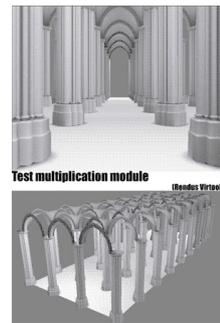


un composant intégrant simplement la colonne



un autre composant constitué du plafond et du sol

Multiplié correctement, ce module permet de générer assez simplement une bonne partie du décor. En plus, en utilisant dans 3dsMax le clonage par référence de l'élément de base, l'équipe s'assure que seul un objet en mémoire est nécessaire. C'est une belle optimisation.

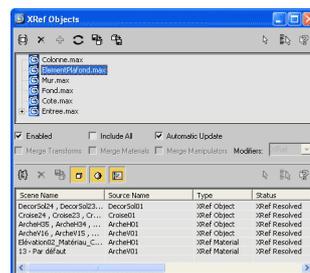


Multiplication du module

Le résultat est globalement satisfaisant mais il apparaît évident que la scène va devenir très rapidement extrêmement complexe. En effet, beaucoup d'autres éléments vont venir se greffer sur la scène déjà existante.

L'équipe des graphistes choisit donc encore une fois de « diviser pour régner » pour sa gestion des fichiers. Chacun des éléments de base (ou module) de notre scène va être placé dans un fichier séparé. Un fichier vide de toute géométrie propre va être créé : le fichier « Cathédrale.max » qui va contenir des références vers les modules.

D'un point de vue technique, on va faire usage des « Xref Objects » pour pointer les modules dont on a besoin.



Fenêtre Xref

Cette solution est très intéressante à plus d'un titre. En premier lieu, l'usage de références l'optimise fortement et permet une maintenance aisée. Mais le plus important est que grâce à l'usage de « XRef objects » plusieurs personnes peuvent intervenir en même temps sur le décor. Un graphiste peut retravailler la colonne pendant qu'un autre s'attache à détailler un peu plus les murs latéraux.

De cette manière les différents modules ont pu être réalisés rapidement : la charge de travail a été répartie entre les différents graphistes, et on a su éviter un goulot d'étranglement.



Module entrée



Module nef



Module côté

Dans « Cathédrale.max », une fois que les modules de base sont multipliés par référence correctement et positionnés on obtient le résultat suivant :



Cathédrale assemblée

La scène est ensuite exportée dans les ressources du projet au format Virtools « Cathe.nmo ». Notre décor apparaîtra bien comme un composant unique alors même qu'il est l'assemblage de différents composants.

4.1.3. Intégration

Pendant que les graphistes travaillaient, les programmeurs ont mis en place une composition Virtools (.cmo) qui contient un petit script qui permet l'intégration automatique. La composition ne contient en fait que deux informations : les composants à charger et comment les charger.

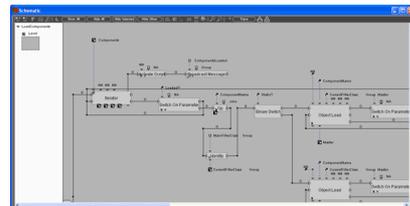
Name	ComponentName	1: Loaded?	2: Run?	3: Log
0	Colonne		FALSE	
1	FxFire		FALSE	
2	Nave		FALSE	
3	AbsFreeCam		FALSE	
4	Interface		FALSE	

Tableau des composants à charger

En outre, les programmeurs ont réalisé les composants supplémentaires suivants :

- « **Statistics** » va permettre d'afficher les statistiques courantes (nombre de faces affichées, fréquence de rafraîchissement) ;
- « **AbsFreeCam** » est la caméra de base qui permet la navigation. C'est une adaptation de la « FreeCam » de base fournie avec Virtools. On a simplement modifié l'affectation des touches de navigation ;
- « **Interface** » est un petit module qui permet d'afficher et de masquer le calque « Fil de fer » et les textures. Il a été réalisé pour permettre de saisir la complexité de la modélisation ;
- enfin « **FxFire** » est un petit effet spécial simulant du feu dans la cheminée. Comme le projet avançait bien, il a été réalisé pour mettre un peu de vie dans ce décor, encore un peu vide.

Chaque composant est stocké dans le répertoire des ressources, avec son aide en html et un fichier expliquant son utilisation, de la même manière que comme les composants par défaut de Virtools. Ainsi la documentation est assurée.



Script de chargement de composants

4.1.4. Tests

Enfin, la première compilation du projet est réalisée. Après les ajustements nécessaires, une version satisfaisante est obtenue.



Tous les composants intégrés dans Virtools

L'équipe publie cette version directement dans une page de Test sur le site du projet : [Lien 256](#). En effet la technologie employée permet une intégration directement en page web, alors autant en profiter.

L'équipe n'attend plus que les commentaires des testeurs pour ajuster ses prochaines orientations. Et d'ores et déjà, on pense à mettre en place un système de commentaires anonymes pour recueillir directement les impressions des testeurs.

5. Conclusion

Nous avons pu voir comment les graphistes ont appliqué les mesures pour le travail en équipe. Correctement orientés, ils ont pu rapidement réaliser un travail satisfaisant.

Toujours est-il que beaucoup de choses manquent ou vont changer : certaines modélisations vont être corrigées et les textures et les shaders doivent être réalisés.

Le résultat final ne sera obtenu qu'après de multiples itérations successives et tous les membres de l'équipe le savent. En discutant et en faisant en sorte que chacun puisse faire correctement son travail ça devrait passer tranquille.

Enfin le premier commentaire arrive : la petite sœur d'un des membres de l'équipe a écrit un petit mail sympathique :

« Quoi ? Pas de collision ? Vous n'avez jamais entendu parler d'arbre BSP ? »

Bing, c'est ça l'assurance qualité : être capable d'entendre ce que les autres disent sur votre produit, et ce n'est pas toujours facile.

Mais déjà, dans l'ombre de leurs bureaux, les programmeurs fous pensent à une méthode pour uniformiser les communications entre scripts...

6. épilogue



Version actuelle

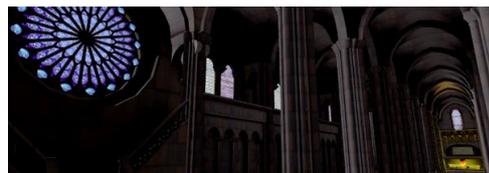
Après un travail sur l'ensemble des composants, le projet a sensiblement évolué pour atteindre une nouvelle version disponible sur cette page : [Lien 257](#). C'est mieux non ?

Pour l'instant le projet reste figé à cette étape. Il n'est pas abandonné mais disons que l'équipe a besoin de faire un break. En fait, il n'y pas d'équipe : le projet a été réalisé par une personne seule, l'auteur du présent article, votre humble serviteur.

À titre indicatif, pour atteindre son niveau actuel (article compris), le projet a demandé près de deux mois et demi à temps plein à une personne expérimentée dans tous les secteurs du développement.

J'espère que ceci vous permettra d'évaluer plus précisément les efforts et délais nécessaires pour votre propre projet, et que les recommandations vous permettront d'obtenir un résultat similaire en moins de temps, ou un résultat de meilleure qualité dans le même temps.

Si vous êtes intéressés pour continuer cette aventure (documentation comprise), contactez-moi par message privé. Et nous pourrions peut-être la fonder, cette équipe.



Retrouvez l'article de Guillaume Lemasson en ligne : [Lien 258](#)

Architecture d'un jeu vidéo 3D

Cet article présentera un modèle d'architecture pour la réalisation de jeux vidéos en 3D. Connaissances requises : C++.

1. Introduction

Je vais présenter un modèle d'architecture pour la réalisation d'un jeu vidéo en 3D. Ce modèle devra pouvoir s'adapter simplement aussi bien à la réalisation d'un jeu à la première personne (FPS), d'un jeu de stratégie (RTS) ou même à d'autres types de jeux. L'architecture présentée pourra également gérer les jeux en mode multijoueurs en réseau local ou sur internet. De même, l'architecture pourra être simplement étendue pour la réalisation de jeux massivement multijoueurs: MMOFPS, MMORTS, MMORPG ...

Cette architecture sera entièrement codée avec des bibliothèques multiplateformes.

L'objectif n'étant pas de fournir un jeu commercialisable, mais un modèle d'architecture sur lequel se baser pour réaliser un jeu.

1.1. Schéma général

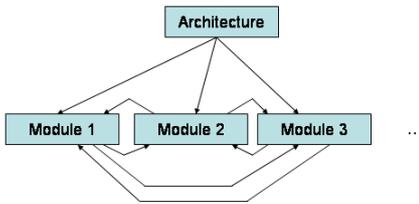
Même sans savoir quel type de jeu on va écrire, il y aura toujours un socle commun, permettant de faire dialoguer des modules, permettant d'interfacer les différents moteurs mis en jeu.

Je vais utiliser les outils suivants :

- Irrlicht pour le rendu 3D la gestion des entrées utilisateur ;
- CEGUI pour la gestion des menus et des éléments graphiques non 3D ;
- asio pour la gestion du réseau et les threads ;
- Irrklang pour le son ;
- boost.

Le jeu contiendra des modules plus ou moins indépendants, gérant chacun un aspect du jeu. L'objectif

étant bien sûr de rendre possibles les développements des différents modules en parallèle, sans que ça n'entraîne d'effets de bord dans les autres modules.

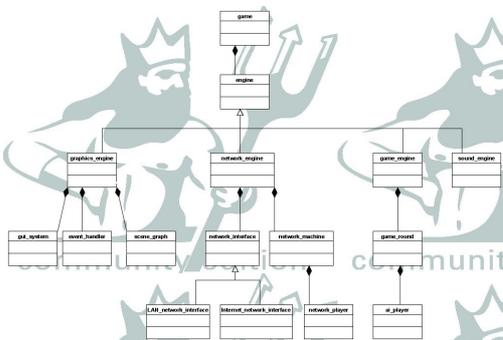


Un objet contiendra tous les modules. D'un point de vue purement conceptuel, l'architecture peut contenir un nombre non défini de modules, il faut donc les stocker dans un conteneur style `std::list`. Chaque module aurait un pointeur vers l'architecture, permettant ainsi d'accéder aux autres éléments de la liste des modules.

1.2. Les contraintes, les objectifs

- L'architecture devra pouvoir convenir à un maximum de types de jeu, il ne faut donc faire aucune supposition sur l'utilisation qui en sera faite.
- Nous avons une très forte contrainte de performances : l'architecture proposée devra être aussi légère que possible tant en ressources processeur, qu'en mémoire ou en espace disque nécessaire.
- L'architecture devra aussi être légère en utilisation du réseau, de manière à favoriser des échanges rapides et supportés par les bandes passantes les plus modestes.
- Les modules étant les pièces de base de l'architecture, ils devront pouvoir communiquer simplement et rapidement entre eux.
- Dans le cas d'un jeu multijoueur, il faudra pouvoir tirer profit des machines puissantes pour leur assigner des tâches telles que la gestion de personnages non joueurs.

Voici un diagramme de classes simplifié pour les principales classes mises en jeu :



2. Les différents modules

Voici la classe de base de notre architecture, celle qui sera instanciée pour démarrer le jeu.

classe de base pour notre architecture

```
class game{
public:
    game();
```

```
~game();
// ...

private:
    std::list<engine*> l_modules;
};
```

Il va nous falloir une classe générique pour représenter un moteur du jeu :

classe générique pour un moteur

```
class engine{
public:
    engine(game*);
    virtual ~engine();

    // accepter les messages des autres moteurs
    void push_event(engine_event& e){
        events_queue.push(e);
    }

    // traite la file des messages
    void process_queue(){
        while (! events_queue.empty()){
            engine_event e =
events_queue.front();
            events_queue.pop();

            process_event(e);
        }

        // traitement propre à chaque moteur
        virtual void frame() = 0;
protected:
    // pointeur vers l'objet contenant
    game *parent;

    // file des messages à traiter
    std::queue<engine_event> events_queue;

    // traitement d'un message, propre à
chaque moteur
    virtual void process_event(engine_event&)
= 0;
};
```

Je détaillerai plus loin le principe des objets `engine_event`, messages envoyés d'un module à l'autre. La classe `engine` possède des fonctions virtuelles pures, elle ne peut donc pas être instanciée. Néanmoins, on remarque que pour pouvoir communiquer avec un autre module, il va falloir remonter à l'objet contenant pour récupérer un pointeur sur le module à atteindre, éventuellement downcaster le pointeur pour pouvoir accéder aux fonctions propres à ce module et ensuite spécifier les opérations à effectuer.

Les files de messages peuvent potentiellement être accédées par des traitements parallèles, chaque moteur doit donc posséder un mutex à verrouiller à chaque ajout de message et à chaque vidage de la liste.

J'ai choisi de mettre les modules suivants :

- le moteur de jeu, pour gérer les différents joueurs et toutes les données propres au jeu ;
- le moteur graphique, chargé de représenter la

- vision du jeu pour un joueur (le joueur ayant physiquement démarré le jeu ;
- le moteur audio, chargé de représenter l'aspect audio du jeu pour un joueur particulier ;
- le moteur de réseau, chargé d'échanger les données avec les autres joueurs.

Le constructeur de la classe engine sert principalement à renseigner le conteneur parent pour chaque module. Ce constructeur devra être appelé à partir des constructeurs des classes filles.

L'exécution proprement dite du jeu consistera simplement en des appels continus à toutes les fonctions frame() des différents moteurs. La condition d'arrêt de cette boucle quasi infinie devra être modifiée par le moteur de jeu. L'architecture possède donc un booléen représentant l'arrêt du programme. Ce booléen devra être protégé par des verrous pour se garantir des accès concurrents éventuels.

```
class game{
    // ...
    boost::mutex still_running_mutex;
    bool still_running;
    // ...
};
```

```
void game::run() {
    bool current_still_running =
still_running;

    // on crée un verrou pour accéder au
mutex
    boost::mutex::scoped_lock
l(still_running_mutex);
l.unlock();

    while (current_still_running) {
        n->frame();
        g->frame();
        gfx->frame();
        s->frame();
        l.lock();

        current_still_running =
still_running;
        l.unlock();
    }
}
```

Ainsi, pour terminer le jeu, il suffira d'appeler un modificateur du booléen still_running.

```
void game::stooooooooo() {
    boost::mutex::scoped_lock
l(still_running_mutex);
    still_running = false;
}
```

La libération du mutex est faite automatiquement à la fin de la portée du verrou. Le destructeur de boost::mutex::scoped_lock appelle la fonction unlock(), il n'est donc pas nécessaire de le spécifier explicitement.

Le destructeur de la classe game va appeler les destructeurs de tous les modules instanciés.

```
game::~game() {
```

```
delete n;
delete g;
delete gfx;
delete s;
}
```

2.1. Communication entre les modules

La communication entre les modules étant vraiment à la base de notre architecture, de très nombreux appels seront effectués pour faire communiquer nos modules, cet aspect doit donc être le plus rapide possible. On pourrait mettre en place un système d'accessoir dans l'architecture pour renvoyer les pointeurs demandés, mais ça restera beaucoup plus lourd que des accès directs. C'est pourquoi j'ai choisi de donner à chaque moteur des pointeurs 'en dur' vers les autres modules. Cela nécessite de connaître à l'avance le nombre de modules à mettre en jeu.

Chaque moteur comporte ainsi quatre pointeurs vers les moteurs de jeu, graphique, audio et réseau. De même, chaque module possède une fonction permettant d'atteindre directement les files de messages des autres modules. Pour cela, il est nécessaire de lier tous les modules entre eux lors de leur création.

```
class game{
    // ...
    game_engine *g;
    graphics_engine *gfx;
    network_engine *n;
    sound_engine *s;
};
```

```
class engine{
    // ...
    void
send_message_to_graphics(engine_event&);
    void
send_message_to_network(engine_event&);
    void send_message_to_game(engine_event&);
    void
send_message_to_sound(engine_event&);

    game_engine *ge;
    network_engine *ne;
    graphics_engine *gfxe;
    sound_engine *se;

    // ...
};
```

Voici le code du constructeur de l'architecture, qui va lier tous les modules entre eux :

```
game::game() {
    n = new network_engine(this);
    g = new game_engine(this);
    gfx = new graphics_engine(this);
    s = new sound_engine(this);

    // lier tous les modules ensemble
    n->attach_game_engine(g);
    n->attach_graphics_engine(gfx);
    n->attach_sound_engine(s);

    g->attach_graphics_engine(gfx);
    g->attach_network_engine(n);
```

```

g->attach_sound_engine(s);

gfx->attach_game_engine(g);
gfx->attach_network_engine(n);
gfx->attach_sound_engine(s);

s->attach_game_engine(g);
s->attach_graphics_engine(gfx);
s->attach_network_engine(n);
}

```

Les différentes fonctions attach_XXX se contentent de recopier les adresses fournies dans les pointeurs vers les autres modules. Une fois l'architecture créée, tous les modules sont capables de communiquer entre eux, c'est déjà une bonne chose de faite.

Les modules ont donc deux moyens pour communiquer entre eux : s'envoyer un message au travers des files de messages ou bien appeler directement les fonctions des autres modules. L'envoi de messages a l'avantage qu'il renforce la séparation des modules, par contre, pour chaque message qu'on peut envoyer d'un module à l'autre, le module récepteur devra être capable de le lire et de l'interpréter. L'envoi de messages est légèrement plus coûteux puisqu'il va demander l'empilage du message sur la file du récepteur ainsi que l'analyse du message lors de sa réception. Il ne faudra donc pas trop abuser des envois de messages entre les modules pour privilégier les appels directs.

2.1.1. Les messages

Il nous faut une classe pouvant représenter tous les types de messages qui peuvent être échangés entre nos modules. Et pour faire encore plus générique, il faudrait que cette classe puisse aussi être utilisée pour communiquer entre les différentes machines du réseau. Les différents modules et les différentes machines peuvent s'envoyer des informations de diverses natures : des chaînes de caractères, des simples nombres, des vecteurs 3D représentant des positions...

J'ai choisi d'utiliser une classe comprenant des std::map pour différents types de données :

```

class engine_event{
public:
    int type;
    std::map<std::string, std::string>
s_data;
    std::map<std::string, int> i_data;
    std::map<std::string, float> f_data;
    std::map<std::string,
serializable_vector3df> v_data;
    bool operator==(const engine_event& e);
    template<class Archive>
    void serialize(Archive& ar, const
unsigned int);
};

```

serializable_vector3df représente simplement une classe fille de la classe de vecteur 3D de flottants d'Irrlicht. J'ai spécialisé cette classe pour la rendre sérialisable, donc pour pouvoir l'envoyer simplement sur le réseau en utilisant boost::serialize :

classe de vecteurs sérialisable

```

class serializable_vector3df : public
irr::core::vector3df{
public:
    serializable_vector3df(){}
    serializable_vector3df(irr::core::vector3
df& v):irr::core::vector3df(v){}
    template<class Archive>
    void serialize(Archive& ar, const
unsigned int){
        ar & X;
        ar & Y;
        ar & Z;
    }
};

```

L'attribut engine_event::type permettra de repérer le type de message, de manière à orienter la recherche des informations dans les std::map.

Il faudra penser à utiliser des clefs de std::map aussi courtes que possible : en effet, ces clefs seront elles aussi sérialisées et envoyées. Quitte même à les troquer contre des int ou d'autres types plus légers.

Suivant les informations que vous choisirez d'envoyer par l'intermédiaire de ces engine_events, vous n'utiliserez pas nécessairement les quatre std::maps proposées, vous pourrez donc sans problème en supprimer l'une ou l'autre. Les messages s'en trouveront plus légers et plus rapides à échanger, notamment sur le réseau.

2.2. Ajout d'un module

Dans le cas de l'utilisation d'une liste de modules std::list<engine*> il est très simple d'ajouter un module, il suffit d'ajouter un item à cette liste. Par contre dans notre cas, où les modules accèdent directement les uns aux autres, il va falloir rajouter un pointeur membre dans la classe game ainsi que dans la classe engine. De même, il va falloir attacher ce nouveau module à tous ceux déjà existants.

2.3. Utilisation des modules en mode passif

Toujours dans un souci de réutilisation, certains modules doivent pouvoir être démarrés en mode *passif*. Par exemple, il pourra être intéressant de ne pas faire d'affichage dans le moteur graphique dans le cas d'un serveur de jeu. On gagnera en performances. En effet, le moteur graphique et le moteur audio ne sont que la partie visible du jeu pour un joueur donné. L'idée est de pouvoir lancer le jeu en mode démon, pour qu'il puisse se concentrer sur la gestion du jeu et des autres joueurs.

Pour pouvoir lancer le jeu dans un tel mode, il va falloir paramétrer l'exécution : en passant des paramètres par la ligne de commande.

J'ai donc créé une petite classe permettant d'analyser les paramètres de la ligne de commande. Cette classe n'est pas primordiale, elle a plus un rôle d'accessoire qu'autre chose, mais elle est bien pratique tout de même.

```

class parameter_analyser{
public:
    parameter_analyser();
};

```

```

    parameter_analyser(int,
char**); // argc, argv
    std::map<std::string, bool> data;
};

```

Le constructeur se contente d'analyser tous les paramètres de la ligne de commande et de remplir la `std::map` à true lorsque le paramètre a été spécifié. Si les paramètres ne sont pas spécifiés dans la ligne de commande, les valeurs de `data` sont mises à false. Par exemple, une exécution avec l'option `--daemon` ou `-d` mettra la valeur associée à la clef "daemon" à true.

Il suffit ensuite de passer une référence sur l'instance de `parameter_analyser` au constructeur `game()` qui analysera les données pour appeler un constructeur de module ou un autre. Ainsi, on peut créer un constructeur de moteur graphique qui accepte un booléen, suivant qu'il faut ou non créer un contexte graphique. Idem pour le moteur audio.

Le jeu démarré en mode passif doit être contrôlé différemment. En effet, vous ne pouvez pas interagir avec la souris ni le clavier, et c'est encore plus vrai si vous associez au mode passif la création d'un nouveau processus de manière à rendre la main sitôt après le démarrage du jeu. C'est la raison pour laquelle il va falloir mettre en place un système de console connaissant quelques commandes de base.

Prenons le cas d'un serveur de jeu, il faut pouvoir communiquer avec le serveur via le réseau. Notre console va donc presque naturellement écouter le réseau, accepter les connexions sur un port particulier, éventuellement faire une authentification par login / mot de passe puis va être en écoute des commandes. Pour sécuriser encore la transmission, on pourrait passer par une couche SSL ([Lien 259](#)).

```

class console{
public:
    console();
    ~console();
    void process_command(std::string&);

    // fonction à lancer pour chaque client
    void
server_thread_tcp_receive(asio::ip::tcp::socket
*);
    // ...
private:
    // attributs propres à la gestion
réseau : sockets, contexte réseau...
};

```

La réalisation d'une telle console revient ni plus ni moins à implémenter un serveur multiclent. Et pour les mêmes raisons que dans mon précédent article sur une architecture de serveur multithreads, j'ai choisi d'utiliser un contexte multithreadé. Nous pouvons ainsi simplement gérer plusieurs connexions simultanées. Pour des raisons de simplicité et surtout de robustesse, j'ai choisi d'utiliser la bibliothèque asio pour la gestion du réseau.

2.3.1. Implémentation du système de console

Il va falloir rattacher la console quelque part dans notre architecture. J'ai choisi de la placer en attribut du moteur

de jeu. Elle sera donc créée lors de son instantiation.

La déclaration complète de la console

```

class console{
public:
    console(game_engine*);
    ~console();
    void process_command(CEGUI::String&);
    void process_command(std::string&);

    void handle_accept_tcp(const
asio::error_code&, asio::ip::tcp::socket*);
    void
server_thread_tcp_receive(asio::ip::tcp::socket
*);
private:
    game_engine *parent;
    asio::ip::tcp::socket *s;
    asio::ip::tcp::acceptor *tcp_acceptor;
    asio::io_service io;
};

```

Le constructeur

```

console::console(game_engine* g):parent(g) {
    // création de la socket d'écoute
    s = new asio::ip::tcp::socket(io);
    tcp_acceptor = new
asio::ip::tcp::acceptor(io,
asio::ip::tcp::endpoint(asio::ip::tcp::v4(),
12345));

    asio::error_code e;
    tcp_acceptor->async_accept(*s,
boost::bind(&console::handle_accept_tcp, this, e,
s));

    asio::thread
t(boost::bind(&asio::io_service::run, &io));

    if (e.value() != 0){
        std::cerr << e.message() <<
std::endl;
    }
}

```

Le constructeur crée une socket qui va écouter les connexions TCP sur le port 12345. A chaque connexion entrante, il va appeler `console::handle_accept_tcp`. Cet appel est effectué en mode asynchrone. Chaque appel à `handle_accept_tcp` va démarrer un thread pour s'occuper de cette connexion :

```

void console::handle_accept_tcp(const
asio::error_code& e, asio::ip::tcp::socket*
socket){
    if (e.value() != 0){
        std::cerr << e.message() <<
std::endl;
        return;
    }

    // on démarre le thread du client
    asio::thread
t(boost::bind(&console::server_thread_tcp_receive
, this, socket));

    // on réarme l'appel asynchrone avec une
nouvelle socket
    asio::ip::tcp::socket *s = new
asio::ip::tcp::socket(io);
}

```

```
asio::error_code ec;
tcp_acceptor->async_accept(*s,
boost::bind(&console::handle_accept_tcp, this,
ec, s));
}
```

Il ne nous reste plus qu'à écouter les instructions envoyées par le client :

```
void console::server_thread_tcp_receive
(asio::ip::tcp::socket *s){
    // authentification éventuelle

    // on attend les instructions de manière
    bloquante
    for (;;) {
        boost::array<char, 1024> buf;
        asio::error_code error;

        size_t len = s-
>read_some(asio::buffer(buf), error);

        if (error == asio::error::eof)
            break; // la connexion a
été interrompue
        else if (error)
            break;

        process_command(std::string(buf.d
ata()));
    }
    s->close();
    delete s;
}
```

Le protocole est extrêmement simple : le client envoie des instructions et le serveur lui renvoie le résultat de la réception, tout ça sous forme de chaînes de caractères.

Et à chaque réception d'une instruction, on appelle le traitement de cette instruction. C'est ici qu'on doit implémenter au moins la commande de fermeture.

```
void console::process_command(std::string& c){
    if (c == "quit")
        parent->parent->stooooooooop();

    // autres instructions existantes
}
```

2.3.2. Un client pour la console

J'ai écrit un client basique pour pouvoir accéder à la console à distance sur un tel serveur de jeu. Il s'agit simplement d'une fenêtre de saisie de commande :



Le code source de ce client est également disponible dans l'archive en téléchargement à la fin de cet article.

Retrouvez la suite de l'article de Pierre Schwartz en ligne : [Lien 260](#)

Liens

- Lien 01 : <http://dico.developpez.com/html/910-Langages-JSP-Java-Server-Pages.php>
- Lien 02 : <http://java.developpez.com/cours>
- Lien 03 : <http://java.developpez.com/>
- Lien 04 : <http://bruce-eckel.developpez.com/livres/java/traduction/tij2/>
- Lien 05 : <http://jmdoudoux.developpez.com/java/>
- Lien 06 : <http://mbaron.developpez.com/javase/java/>
- Lien 07 : <http://java.developpez.com/livres/?page=Debutants#L2212111002>
- Lien 08 : <http://java.developpez.com/livres/?page=Debutants#L2212117485>
- Lien 09 : <http://java.developpez.com/livres/?page=Debutants#L284177371X>
- Lien 10 : <http://dico.developpez.com/html/2990-Langages-EDI-Environnement-de-Developpement-Integre.php>
- Lien 11 : <http://eclipse.developpez.com/>
- Lien 12 : <http://netbeans.developpez.com/>
- Lien 13 : <http://dico.developpez.com/html/749-Langages-AWT-Abstract-Window-Toolkit.php>
- Lien 14 : <http://dico.developpez.com/html/1031-Langages-Swing.php>
- Lien 15 : <http://baptiste-wicht.developpez.com/tutoriel/java/swing/debutant/>
- Lien 16 : <http://gfx.developpez.com/tutoriel/java/gui/>
- Lien 17 : <http://gfx.developpez.com/tutoriel/java/swt/>
- Lien 18 : <http://java.developpez.com/livres/?page=Desktop#L2100081799>
- Lien 19 : <http://java.developpez.com/livres/?page=Francais#L2212120192>
- Lien 20 : <http://dico.developpez.com/html/156-Langages-JDBC-Java-DataBase-Connectivity.php>
- Lien 21 : <http://dico.developpez.com/html/3039-Langages-Hibernate.php>
- Lien 22 : <http://java.developpez.com/IntroJDBC.pdf>
- Lien 23 : <http://jguillard.developpez.com/JDBC/>
- Lien 24 : <http://gfx.developpez.com/tutoriel/java/hibernate/>
- Lien 25 : <http://tahe.developpez.com/java/jpa/>
- Lien 26 : <http://java.developpez.com/livres/?page=Francais#L2212116446>
- Lien 27 : <http://ibatis.apache.org/>
- Lien 28 : <http://dico.developpez.com/html/155-Langages-J2EE-Java-2-Entreprise-Edition.php>
- Lien 29 : <http://dico.developpez.com/html/910-Langages-JSP-Java-Server-Pages.php>
- Lien 30 : <http://dico.developpez.com/html/903-Langages-servlet.php>
- Lien 31 : <http://dico.developpez.com/html/748-Langages-Applet.php>
- Lien 32 : <http://dico.developpez.com/html/3108-Langages-Tomcat.php>
- Lien 33 : <http://adiguba.developpez.com/tutoriels/j2ee/jsp/jstl/>
- Lien 34 : <http://tahe.developpez.com/java/struts/>
- Lien 35 : <http://java.developpez.com/livres/?page=JavaFE#L221211940>
- Lien 36 : <http://java.developpez.com/livres/?page=Web#L2212112319>
- Lien 37 : <http://dico.developpez.com/html/158-Langages-J2ME-Java-2-Micro-Edition.php>
- Lien 38 : <http://default.developpez.com/tutoriel/java/j2me/>
- Lien 39 : <http://java.sun.com/j2me/docs/index.html>
- Lien 40 : <http://duj.developpez.com/tutoriels/java/dessin/intro/>
- Lien 41 : <http://info-rital.developpez.com/tutoriel/java/opengl/opengl1/>
- Lien 42 : <http://mari.developpez.com/tutoriel/java/3d/chap1/>
- Lien 43 : <http://info-rital.developpez.com/tutoriel/java/api/3d/>
- Lien 44 : <http://java.developpez.com/outils/edi/>
- Lien 45 : <http://baptiste-wicht.developpez.com/tutoriel/java/executables/>
- Lien 46 : <http://java.developpez.com/articles/ant/>
- Lien 47 : <http://java.developpez.com/faq/maven/>
- Lien 48 : <http://sebastien-estienne.developpez.com/tutoriels/java/>
- Lien 49 : <http://java.developpez.com/faq/java/>
- Lien 50 : <http://java.developpez.com/faq/gui/>
- Lien 51 : <http://java.developpez.com/faq/xml/>
- Lien 52 : <http://java.developpez.com/faq/javaee>
- Lien 53 : <http://java.developpez.com/faq/javame>
- Lien 54 : <http://java.developpez.com/faq/jdbc>
- Lien 55 : <http://java.developpez.com/faq/hibernate>
- Lien 56 : <http://java.developpez.com/faq/struts>
- Lien 57 : <http://java.developpez.com/faq/eclipse>
- Lien 58 : <http://java.developpez.com/faq/netbeans>
- Lien 59 : <http://java.developpez.com/faq/jcreator>
- Lien 60 : <http://java.developpez.com/faq/maven>
- Lien 61 : <http://www.developpez.net/forums/forumdisplay.php?f=22>
- Lien 62 : <http://www.developpez.net/forums/forumdisplay.php?f=203>
- Lien 63 : <http://www.developpez.net/forums/forumdisplay.php?f=193>
- Lien 64 : <http://www.developpez.net/forums/forumdisplay.php?f=295>
- Lien 65 : <http://www.developpez.net/forums/forumdisplay.php?f=735>
- Lien 66 : <http://www.developpez.net/forums/forumdisplay.php?f=306>
- Lien 67 : <http://www.developpez.net/forums/forumdisplay.php?f=194>
- Lien 68 : <http://www.developpez.net/forums/forumdisplay.php?f=213>
- Lien 69 : <http://www.developpez.net/forums/forumdisplay.php?f=114>
- Lien 70 : <http://www.developpez.net/forums/forumdisplay.php?f=200>
- Lien 71 : <http://www.developpez.net/forums/forumdisplay.php?f=201>
- Lien 72 : <http://www.developpez.net/forums/forumdisplay.php?f=629>
- Lien 73 : <http://www.developpez.net/forums/forumdisplay.php?f=296>
- Lien 74 : <http://www.developpez.net/forums/forumdisplay.php?f=303>
- Lien 75 : <http://www.developpez.net/forums/forumdisplay.php?f=199>
- Lien 76 : <http://baptiste-wicht.developpez.com/tutoriels/java/debuter/>
- Lien 77 : <http://developer.android.com/sdk/index.html>

Lien 78 : <http://www.eclipse.org/downloads/>
Lien 79 : <http://java.sun.com/javase/downloads/index.jsp>
Lien 80 : <https://dl-ssl.google.com/android/eclipse/>
Lien 81 : <http://developer.android.comLien/>
Lien 82 : <http://developer.android.coLien/>
Lien 83 : <http://developer.android.com/reference/android/app/Activity.html>
Lien 84 : <http://nbenbourahla.developpez.com/tutoriels/android/introduction-programmation-android/>
Lien 85 : <http://www.dil.univ-mrs.fr/~garreta/javaGen/eclipse/index.html>
Lien 86 : <http://www.dil.univ-mrs.fr/~garreta/>
Lien 87 : <http://www.univmed.fr/>
Lien 88 : <http://www.sciences.univmed.fr/>
Lien 89 : <http://java.sun.com/>
Lien 90 : <http://java.sun.com/javase/6/webnotes/install/jdk/install-windows.html>
Lien 91 : <http://java.sun.com/javase/6/webnotes/install/jdk/install-linux.html>
Lien 92 : <http://java.sun.com/javase/6/docs/api/>
Lien 93 : <http://www.eclipse.org/>
Lien 94 : <http://www.eclipseplugincentral.com/>
Lien 95 : <http://www.eclipse-plugins.2y.net/>
Lien 96 : <http://www.eclipsetotale.com/>
Lien 97 : <http://henri-garreta.developpez.com/tutoriels/eclipse/installation-utilisation-eclipse-developpement-java/#LX>
Lien 98 : <http://henri-garreta.developpez.com/tutoriels/eclipse/installation-utilisation-eclipse-developpement-java/>
Lien 99 : <http://heureuxoli.developpez.com/office/outlook/regles/>
Lien 100 : <http://www.microsoft.com/downloads/details.aspx?FamilyID=220549b5-0b07-4448-8848-dcc397514b41&DisplayLang=fr>
Lien 101 : <http://heureuxoli.developpez.com/office/word/champs/#LIV-A-5>
Lien 102 : <http://heureuxoli.developpez.com/office/word/champs/#LIV-D-3>
Lien 103 : <http://morpheus.developpez.com/sql-server-2005/>
Lien 104 : <http://vb.developpez.com/bidou/recordset-ado/>
Lien 105 : <http://warin.developpez.com/access/dao/>
Lien 106 : <http://heureuxoli.ftp-developpez.com/office/word/publipostage/fichiers/publipostage.zip>
Lien 107 : <http://heureuxoli.developpez.com/publipostage/>
Lien 108 : <http://silkycroad.developpez.com/Excel/PublipostageWordExcel/>
Lien 109 : <http://warin.developpez.com/access/dao/>
Lien 110 : <http://vb.developpez.com/bidou/recordset-ado/>
Lien 111 : <http://heureuxoli.developpez.com/office/word/publipostage/>
Lien 112 : <http://starec.developpez.com/tuto/pptliste/>
Lien 113 : <http://starec.developpez.com/tuto/pptnavigation/>
Lien 114 : <http://heureuxoli.developpez.com/office/openxml/>
Lien 115 : <http://grandfather.developpez.com/articles/openxml/structure/>
Lien 116 : http://www.ecma-international.org/news/TC45_current_work/TC45_available_docs.htm
Lien 117 : <http://office.microsoft.com/fr-fr/help/HA101675271036.aspx>
Lien 118 : <http://www.microsoft.com/downloads/details.aspx?displaylang=framp:&FamilyID=941B3470-3AE9-4AEE-8F43-C6BB74CD1466>
Lien 119 : <http://support.microsoft.com/kb/923505>
Lien 120 : <http://office.microsoft.com/fr-fr/excel/HA101491511036.aspx?pid=CH100648241036>
Lien 121 : <http://silkycroad.developpez.com/excel/tcd/>
Lien 122 : <http://silkycroad.developpez.com/Excel/NouveautesExcel2007/>
Lien 123 : <http://mhubiche.developpez.com/Access/cours/bases/>
Lien 124 : <http://sqlpro.developpez.com/cours/sqlaz/fondements/>
Lien 125 : <http://access.developpez.com/>
Lien 126 : <http://www.developpez.net/forums/f655/logiciels/microsoft-office/access/requetes-sql/>
Lien 127 : <http://mhubiche.developpez.com/Access/tutoJointures/>
Lien 128 : <http://jeannot45.developpez.com/articles/access/creationrequetes1/#remarques>
Lien 129 : <http://jeannot45.ftp-developpez.com/articles/access/creationrequetes1/fichiers/Bouquins.ZIP>
Lien 130 : <http://jeannot45.developpez.com/articles/access/creationrequetes1/>
Lien 131 : <http://cpp.developpez.com/livres/>
Lien 132 : <http://www.amazon.fr/Je-lance-introduction-programmation-1Cédérom/dp/2742936521>
Lien 133 : <http://www.amazon.fr/Programmation-Applications-avec-Bjarne-Stroustrup/dp/274407442X>
Lien 134 : <http://www.amazon.fr/Accelerated-Practical-Programming-Example-Depth/dp/020170353X>
Lien 135 : <http://www.amazon.fr/Primer-4th-Stanley-B-Lippman/dp/0201721481>
Lien 136 : <http://cpp.developpez.com/cours/>
Lien 137 : <http://cpp.developpez.com/cours/cpp/>
Lien 138 : <http://bruce-eckel.developpez.com/livres/cpp/traduction/ticpp2vol1/>
Lien 139 : <http://algo.developpez.com/>
Lien 140 : <http://conception.developpez.com/>
Lien 141 : <http://cpp.developpez.com/outils/>
Lien 142 : <http://www.microsoft.com/Express/vc/>
Lien 143 : <http://www.codeblocks.org/>
Lien 144 : <http://notepad-plus.sourceforge.net/fr/site.htm>
Lien 145 : <http://www.microsoft.com/Express/vc/>
Lien 146 : <http://www.mingw.org/>
Lien 147 : <http://software.intel.com/en-us/intel-compilers/>
Lien 148 : <http://gcc.gnu.org/>
Lien 149 : <http://www.gnu.org/software/gdb/>
Lien 150 : <http://www.stack.nl/%7Edimitri/doxygen/>
Lien 151 : <http://cpp.developpez.com/faq/cpp/>
Lien 152 : <http://www.developpez.net/forums/f19/c-cpp/cpp/>
Lien 153 : <http://cpp.developpez.com/bibliotheques/>
Lien 154 : <http://cpp.developpez.com/cours/?page=bibliotheques>
Lien 155 : <http://www.comeaucomputing.com/techtalk/>
Lien 156 : <http://www.parashift.com/c++-faq-lite/>
Lien 157 : <http://jlecomte.ifrance.com/c++/c++-faq-lite/index-fr.html>
Lien 158 : http://www.research.att.com/%7Ebs/bs_faq.html
Lien 159 : <http://www.research.att.com/%7Ebs/C++.html>

Lien 160 : <http://www.artima.com/cppsource/>
Lien 161 : <http://www.ddj.com/dept/cpp/>
Lien 162 : <http://alp.developpez.com/tutoriels/debuter-cpp/>
Lien 163 : <http://msdn.microsoft.com/en-us/library/hs24szh9%28VS.100%29.aspx>
Lien 164 : <http://www.microsoft.com/express/downloads/>
Lien 165 : <http://apais.developpez.com/tutoriels/c++/visual-cpp-2010-express/>
Lien 166 : <http://qt.developpez.com/doc/4.4/how-to-learn-qt/>
Lien 167 : <ftp://ftp.developpez.com/qt/tutoriels/introduction-qt/fichiers/tutorial.rar>
Lien 168 : <ftp://ftp.developpez.com/qt/tutoriels/introduction-qt/fichiers/tutorial.tar.bz2>
Lien 169 : <http://qt.developpez.com/doc/4.4/mainwindows-application.html>
Lien 170 : <http://www.developpez.net/forums/redirect-to/?redirect=http%3A%2F%2Fqt.nokia.com%2Fdownloads>
Lien 171 : <ftp://ftp.developpez.com/qt/tutoriels/introduction-qt/fichiers/t1.rar>
Lien 172 : <http://qt.developpez.com/doc/4.4/qcoreapplication.html#arguments>
Lien 173 : <http://qt.developpez.com/tutoriels/introduction-qt/>
Lien 174 : <http://developer.qt.nokia.com/>
Lien 175 : http://developer.qt.nokia.com/wiki/PySide_Newbie_Tutorials
Lien 176 : http://developer.qt.nokia.com/wiki/PySideSimplicissimus_Module_1_Prerequisites
Lien 177 : http://developer.qt.nokia.com/wiki/Setting_up_PySide
Lien 178 : http://developer.qt.nokia.com/wiki/Hello_World_in_PySide
Lien 179 : <http://qt-devnet.developpez.com/tutoriels/pyside/simplissimus/1-fermer/>
Lien 180 : <http://qt-devnet.developpez.com/tutoriels/pyside/simplissimus/2-a-propos/>
Lien 181 : <http://qt-devnet.developpez.com/tutoriels/pyside/simplissimus/3-licence/>
Lien 182 : <http://qt-devnet.developpez.com/tutoriels/pyside/simplissimus/4-combiner/>
Lien 183 : <http://qt-devnet.developpez.com/tutoriels/pyside/simplissimus/5-combinaison-alternative/>
Lien 184 : <http://qt-devnet.developpez.com/tutoriels/pyside/simplissimus/6-tout-combiner/>
Lien 185 : <http://qt-devnet.developpez.com/tutoriels/pyside/simplissimus/7-ingenierie/>
Lien 186 : <http://qt.nokia.com/>
Lien 187 : <http://www.python.org/>
Lien 188 : <http://www.pyside.org/>
Lien 189 : <http://qt-devnet.developpez.com/tutoriels/pyside/simplissimus/>
Lien 190 : http://developer.qt.nokia.com/wiki/PySideSimplicissimus_Module_2_CloseButton
Lien 191 : <ftp://ftp.developpez.com/qt-devnet/tutoriels/pyside/simplissimus/1-fermer/fichiers/quitter.ui>
Lien 192 : <http://pypi.python.org/pypi/setuptools>
Lien 193 : <http://tcuvelier.developpez.com/tutoriels/pyqt/introduction-pyqt/>
Lien 194 : <http://riverbankcomputing.co.uk/software/pyqt/intro>
Lien 195 : <http://riverbankcomputing.co.uk/static/Docs/PyQt4/html/qtcore.html>
Lien 196 : <http://riverbankcomputing.co.uk/static/Docs/PyQt4/html/qtgui.html>
Lien 197 : <http://riverbankcomputing.co.uk/static/Docs/PyQt4/html/qt.html>
Lien 198 : <http://www.qtrac.eu/pyqtbook.html>
Lien 199 : <http://www.commandprompt.com/community/pyqt/>
Lien 200 : <http://www.amazon.com/gp/product/0132354160>
Lien 201 : <http://tcuvelier.developpez.com/tutoriels/pyqt/bases/>
Lien 202 : <http://tcuvelier.developpez.com/tutoriels/web-semantique/rdfa/introduction/>
Lien 203 : <http://tcuvelier.developpez.com/tutoriels/web-semantique/html5-microdonnees/introduction/>
Lien 204 : http://en.wikipedia.org/wiki/Semantic_Web
Lien 205 : <http://xmlns.com/foaf/0.1/>
Lien 206 : <http://purl.org/NET/c4dm/event.owl#>
Lien 207 : <http://purl.org/dc/elements/1.1/>
Lien 208 : <http://data.lirmm.fr/ontologies/files/oan/>
Lien 209 : <http://sites.wiwiw.fu-berlin.de/suhl/bizer/d2r-server/config.rdf#>
Lien 210 : <http://www.w3.org/Submission/vcard-rdf/>
Lien 211 : <http://creativecommons.org/ns#>
Lien 212 : <http://jplu.developpez.com/tutoriels/web-semantique/introduction/>
Lien 213 : <http://jena.sourceforge.net/downloads.html>
Lien 214 : http://openjena.org/tutorial/RDF_API/index.html
Lien 215 : <http://www.w3.org/TR/vcard-rdf>
Lien 216 : <http://ftp.developpez.com/web-semantique/tutoriel/jena/introduction-rdf/fichiers/Tutorial01.java>
Lien 217 : <http://ftp.developpez.com/web-semantique/tutoriel/jena/introduction-rdf/fichiers/Tutorial02.java>
Lien 218 : <http://ftp.developpez.com/web-semantique/tutoriel/jena/introduction-rdf/fichiers/Tutorial03.java>
Lien 219 : <http://www.w3.org/2001/sw/RDFCore/>
Lien 220 : <http://www.w3.org/TR/rdf-testcases/#ntriples>
Lien 221 : <http://web-semantique.developpez.com/tutoriels/jena/introduction-rdf/>
Lien 222 : <http://taslimanka.developpez.com/tutoriels/bi/>
Lien 223 : <http://www.developpez.net/forums/f556/hardware-systemes-logiciels/autres-logiciels/erp/>
Lien 224 : <http://solutions-entreprise.developpez.com/solutions-entreprise/cours/#erp>
Lien 225 : <http://solutions-entreprise.developpez.com/solutions-entreprise/cours/#crm>
Lien 226 : <http://www.developpez.net/forums/f38/bases-donnees/business-intelligence/>
Lien 227 : <http://business-intelligence.developpez.com/cours/>
Lien 228 : <http://www.developpez.net/forums/f1217/hardware-systemes-logiciels/autres-logiciels/ged/>
Lien 229 : <http://solutions-entreprise.developpez.com/solutions-entreprise/>
Lien 230 : <http://fbeaulieu.developpez.com/guide/14-objets-2.html>
Lien 231 : <http://laurent-dardenne.developpez.com/articles/metaclasse/>
Lien 232 : <http://hdd34.developpez.com/cours/artpoo/>
Lien 233 : <http://notepad-plus.sourceforge.net/fr/site.htm>
Lien 234 : <http://www.scintilla.org/SciTE.html>
Lien 235 : <http://smultron.sourceforge.net/>
Lien 236 : <http://perl.developpez.com/telecharger/detail/id/2120/Fraise>
Lien 237 : <http://www.perl.com/>
Lien 238 : <http://www.activestate.com/>
Lien 239 : <http://www.activestate.com/activeperl/downloads>
Lien 240 : <http://perl.developpez.com/tutoriel/cours1/>
Lien 241 : <http://perl.developpez.com/tutoriel/cours2/>

- Lien 242 : <http://perl.developpez.com/tutoriel/cours3/>
- Lien 243 : <http://lhullier.developpez.com/tutoriels/perl/intro/>
- Lien 244 : <http://perl.developpez.com/cours/>
- Lien 245 : <http://perl.developpez.com/livres/#L284177404X>
- Lien 246 : <http://perl.developpez.com/livres/#L2841771407>
- Lien 247 : <http://perl.developpez.com/livres/>
- Lien 248 : <http://perl.developpez.com/cours/>
- Lien 249 : <http://perl.developpez.com/faq/>
- Lien 250 : <http://www.developpez.net/forums/forumdisplay.php?f=82>
- Lien 251 : <http://www.perl.org/>
- Lien 252 : <http://perl.enstimac.fr/>
- Lien 253 : <http://www.cpan.org/>
- Lien 254 : <http://woufeil.developpez.com/tutoriels/perl/debuter/>
- Lien 255 : <http://ericreboisson.developpez.com/tutoriels/install-subversion/>
- Lien 256 : <http://lemasson.developpez.com/tutoriels/methodologie/initiation-developpement-en-equipe/vmo/aicBastique.html>
- Lien 257 : <http://lemasson.developpez.com/tutoriels/methodologie/initiation-developpement-en-equipe/vmo/aicAvance.html>
- Lien 258 : <http://lemasson.developpez.com/tutoriels/methodologie/initiation-developpement-en-equipe/>
- Lien 259 : <http://dico.developpez.com/html/1589-Securite-SSL-Secure-Sockets-Layer.php>
- Lien 260 : <http://khayyam.developpez.com/articles/cpp/jeux/architecture/>