



Develloppez

Le Mag

Edition de Juin - Juillet 2009.

Numéro 22.

Magazine en ligne gratuit.

Diffusion de copies conformes à l'original autorisée.

Réalisation : Alexandre Pottiez

Rédaction : la rédaction de Develloppez

Contact : magazine@redaction-develloppez.com

Sommaire

Le Club	Page 2
Java/Eclipse	Page 3
Dev. Web	Page 8
(X)HTML/CSS	Page 14
Flex	Page 15
JavaScript	Page 18
Ruby on Rails	Page 22
AJAX	Page 26
C/C++/GTK/Qt	Page 29
MS Office	Page 36
Hardware	Page 42
Mac	Page 43
Linux	Page 48
Conception	Page 54
2D/3D/Jeux	Page 60
Liens	Page 67

Article Dev. Web



Maîtriser Firebug, l'indispensable extension Firefox pour le développement web

Découvrez les principales caractéristiques de Firebug et comprenez son fonctionnement.

par **Eric Pommereau**
Page 7

Article Ruby on Rails



Commencer avec Ruby on Rails

Découvrez les atouts de ce framework qui ne laisse pas indifférent.

par **Vincent Dubois**
Page 22

Editorial

En même temps que les beaux jours reviens votre magazine préféré.

Retrouvez une sélection de nos meilleurs articles, critiques de livres, et questions/réponses sur diverses technologies.

Profitez-en bien !

La rédaction

Un nouveau portail dynamique à la fois professionnel et communautaire : vos avis ?

Depuis que vous connaissez Developpez, c'est-à-dire depuis ô combien d'années, nous avons toujours eu sensiblement la même page d'accueil pour faire nos annonces.

Toutefois, ce soir marque le début d'une série de changements sur nos "index de rubrique", comme nous les appelons. Nous allons commencer par l'index de la rubrique PHP, en test pendant quelque temps. Pendant cette période bêta, nous aimerions avoir votre avis sur la nouvelle ergonomie que nous avons choisie et sur les nouvelles fonctionnalités. N'hésitez pas à me spammer par MP, à donner à la suite vos impressions et vos suggestions, voire même les bugs si vous en trouvez.

Vous pouvez voir que nos convictions ne changent pas avec le temps, Developpez reste un espace de diffusion d'information gratuite et de qualité, et surtout (pour vous) nous refusons catégoriquement les publicités intrusives. Nous faisons notre possible pour mettre en avant les informations les plus utiles, c'est pourquoi nous avons adopté un système à trois colonnes pour notre nouvelle organisation.

Le grand changement pour vous, nos lecteurs, est que vous pourrez suggérer vos propres annonces. Par exemple, si vous voyez un débat très intéressant sur les forums, vous pouvez en faire la promotion vous-mêmes à l'aide de notre formulaire de suggestion d'annonces (connectez-vous en bas à droite avec votre identifiant forum).

Qu'en pensez-vous ?

Commentez cette news en ligne : [Lien1](#)

Revue de l'IDE JetBrains IntelliJ Idea 8

Voici mon avis sur la nouvelle version de IntelliJ IDEA 8, IDE de la société JetBrains. Une alternative intéressante par rapport à Eclipse ou Netbeans.

1. Introduction

À l'heure où Eclipse et Netbeans règnent sans conteste sur le marché des IDEs Java, des solutions alternatives existent cependant.

C'est le cas par exemple de IntelliJ Idea de l'éditeur JetBrains.

Bien que celui-ci ne soit ni Open Source, ni gratuit, il n'en reste pas moins un excellent outil. Il est très bien fini, stable et supporte la plupart des frameworks Java les plus utilisés.

Bien que principalement orienté Java, il n'est pas limité à celui-ci : Groovy, (J)Ruby, GWT, Javascript, Flex sont supportés.

Son support de Groovy et Grails est par ailleurs le meilleur à l'heure actuel.

L'IDE et ses principales fonctionnalités ayant déjà été présentés par des revues de la version 6 par Baptiste Wicht ([Lien2](#)) ou Valère Déjardin ([Lien3](#)), ainsi que ma propre review ([Lien4](#)) de la version 7, je ne m'attarderai que sur les principales nouveautés de la version 8.



2. Revue

2.1. Installation et configuration

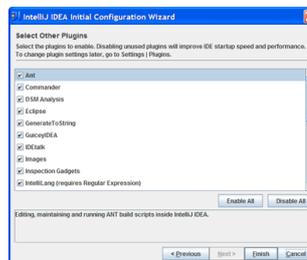
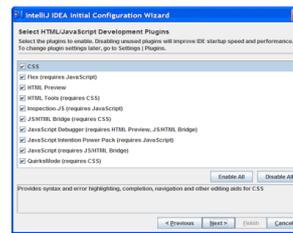
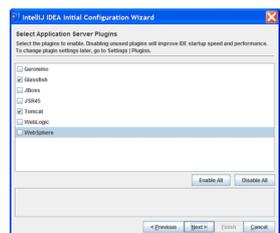
Idea 8 ajoutant encore un nombre important de frameworks, cette nouvelle version permet lors du premier lancement de personnaliser le profil d'utilisation.

Cela se traduit par divers écrans permettant d'activer ou désactiver certains plug-ins.

L'intérêt est double selon moi. Le premier étant simplement d'améliorer les performances et la vitesse de démarrage. Les utilisateurs d'Eclipse connaissent cela, plus

il y a de plug-in, plus c'est gourmand et lent.

Le second intérêt permet aussi au développeur n'ayant pas lu cette review d'avoir une idée des fonctionnalités de l'outil qu'il va utiliser ;-)

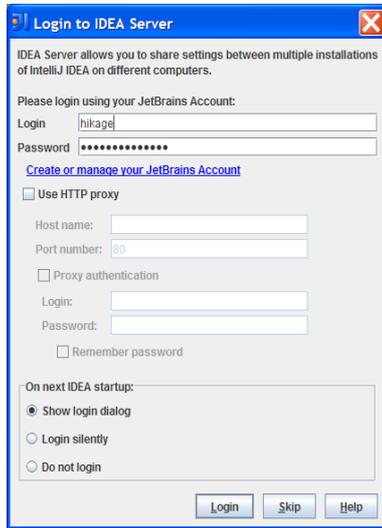


2.2. Partage de configuration

Une autre nouveauté intéressante est le partage de la configuration entre plusieurs installations d'IntelliJ Idea.

Après avoir créer un compte sur le site de JetBrains ([Lien5](#)), votre configuration est automatiquement synchronisée.

Ce qui peut être pratique si vous avez vos propres raccourcis claviers et que vous devez travailler temporairement sur une autre station.



Via un autre mécanisme, il est possible de partager une configuration par équipe, et éviter le problème de différences lors de l'autoformatage du code par exemple (et donc les problèmes de merging CVS/Subversion).

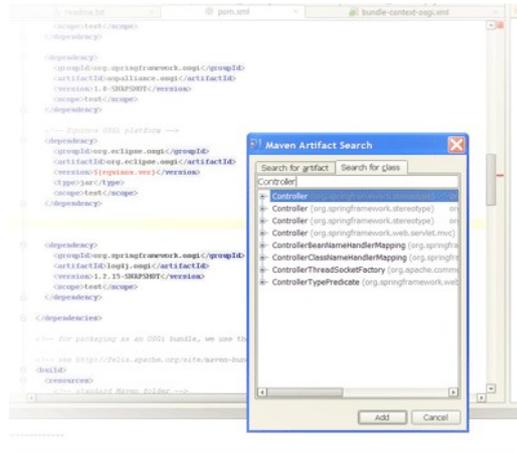
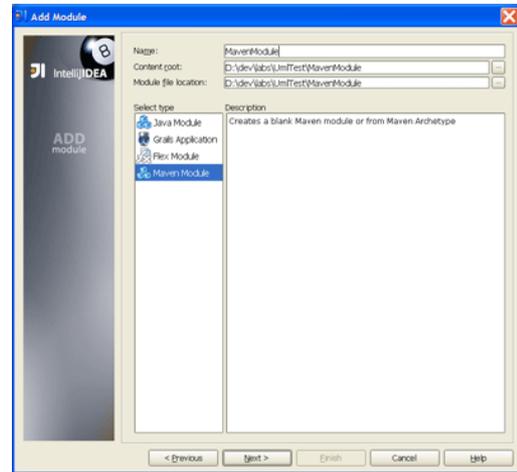
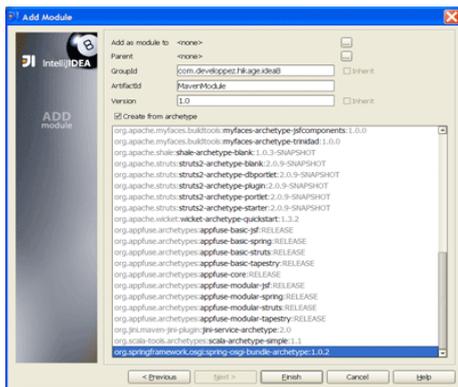
2.3. Amélioration du support de Maven

La version 7 apportait la possibilité d'importer un fichier POM pour créer un projet Idea, la version 8 va plus loin en permettant directement la création d'un projet Maven.

Un assistant permet de définir le nom du groupe, de l'artefact ainsi que sa version, mais aussi de choisir un archétype à utiliser pour créer le template du projet.

A coté de cela, l'éditeur de POM propose une autocomplétion des plug-ins, des raccourcis pour ajouter une dépendance, etc.

Pour information, le bug concernant le lancement de Jetty via le plug-in Maven semble résolu.

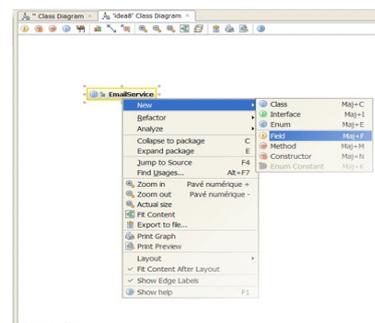
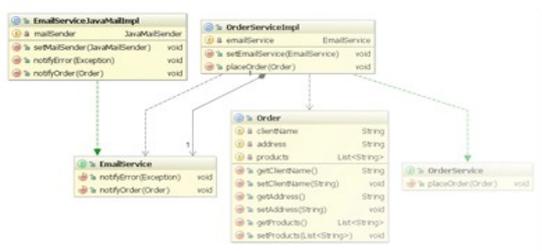


2.4. Editeur de classes UML

Une autre nouveauté dans Idea est l'apparition d'un éditeur de diagramme de classes. Il ne remplacera certainement pas un outil commercial dédié à l'UML mais reste un petit outil pratique.

L'éditeur permet de visualiser un package, d'ajouter une classe, de lui ajouter des attributs, méthodes ou constructeur.

Le tout étant directement synchronisé avec le code Java.



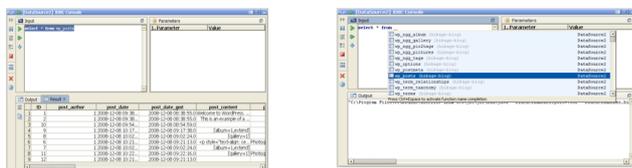
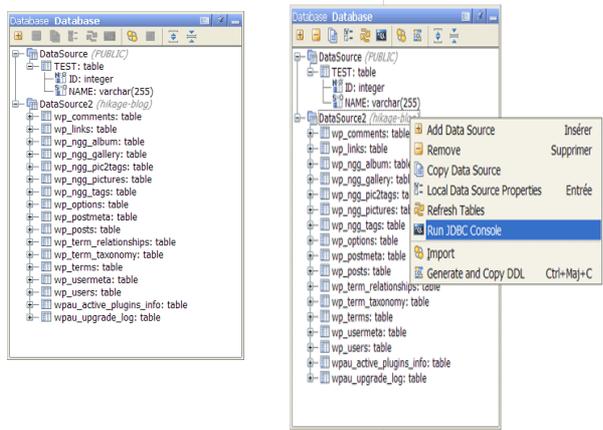
2.5. Console SQL

Un des manques de Idea était de ne pas avoir d'outil pour créer et exécuter des commandes SQL, un outil externe devant dès lors être utilisé.

Cette nouvelle version apporte à la fois un éditeur SQL proposant une complétion pour le SQL Standard mais aussi pour des dialectes spécifiques tels que Oracle, Mysql ou SQLLite.

Une fois le script créé, il sera possible de lancer celui-ci et de visualiser le résultat.

Une fonctionnalité intéressante concernant l'autocomplétion des noms de tables est qu'Idea peut se baser sur une base de données existante en effectuant l'analyse des métadonnées. Mais il peut aussi se baser sur un script de création de tables et donc sur une base de données non existante.



2.6. Support de nouveaux frameworks

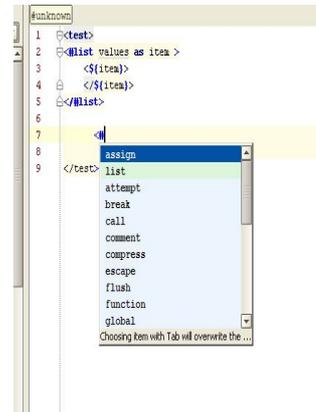
Traditionnellement, les nouvelles versions de Idea apportent le support de nouveaux frameworks ou outils à la mode. Cette release ne déroge pas à la règle.

Au niveau Web, citons par exemple Struts 2, JBoss Seam ou encore GWT 1.5.

N'ayant jamais utilisé ces framework, je ne saurais juger de la qualité de l'intégration dans l'IDE. Mais d'après ce que je sais, on peut compter sur des aides à la configuration, des outils de vérification, ou encore un éditeur visuel de pageflow pour JBoss Seam.

Le support des services REST et de la JRS-311 est aussi un apport de cette version.

Les deux moteurs de template Java, Velocity et Freemarker, sont aussi gérés avec un éditeur doté d'une autocomplétion.



2.7. Amélioration des supports existants

Concernant Spring, cette nouvelle version gère correctement Spring 2.5 et en particulier les annotations (@Component, @Service, @Repository, @Qualifier). Le support de Spring DM est aussi une nouveauté, ainsi que de Spring WebFlow (et de son éditeur de flow visuel).

Le support de Grails se voit amélioré par diverses nouveautés. Une particulièrement pratique est l'autocomplétion des attributs dans les Tags Grails. Avant il était nécessaire de lire la documentation ou le code pour connaître les divers attributs possibles.

Les plug-ins peuvent maintenant être directement intégrés via l'IDE, et ne nécessite plus l'utilisation de la ligne de commande.

2.8. Petits outils sympathiques

D'anciens petits plug-ins utiles ont été directement intégrés dans l'IDE.

Le premier sera particulièrement utile aux développeurs de produits OpenSource car il permet d'ajouter ou remplacer le texte de la licence dans les fichiers.

Exit les copier/coller et les vérifications manuelles !

Un deuxième exemple est l'intégration d'un évaluateur XPath. Pratique aussi pour ceux qui ne disposent pas d'outil comme XMLSpy et qui s'arrachent les cheveux avec une expression XPath !

2.9. En Vrac

Voici une liste d'autres fonctionnalités ou supports ajoutés dans cette version que je n'ai pas la possibilité de tester par moi même:

- Support de Git
- Support de Team Foundation Server
- Support des job de Peforce
- Debugger Flex

3. Conclusion

Idea 8 reste fidèle à la qualité des précédentes versions, le support des nouveaux frameworks particulièrement bien choisis et les nouveaux petits outils intégrés souvent des plus utiles.

Idea reste selon moi une alternative intéressante pour le

développement Java. Le seul problème est son prix, qui en rebute plus d'un vers Eclipse ou Netbeans qui sont aussi de très bonne facture.

Par contre, pour ce qui est développement Groovy et Grails, en attendant de voir le travail de SpringSource pour un support de qualité dans Eclipse, Idea reste clairement la

référence, et permet à cet IDE de se démarquer véritablement des autres.

Retrouvez l'article de Gildas Cuisinier en ligne : [Lien6](#)
Suivez l'évolution de la prochaine version d'IntelliJ IDEA et téléchargez les previews : [Lien103](#)

Les dernières news

Nouvel espace dédié à Spring sur Developpez.com

Devant le succès du framework Spring et de l'ensemble du portfolio Spring ainsi que la dynamique et l'enthousiasme de la communauté, nous avons mis en place un espace entièrement dédié à la communauté francophone autour de Spring : plus de visibilité, une meilleure navigation à travers les ressources consacrées à Spring, et bientôt davantage d'ouverture avec la nouvelle version du portail.

Cette nouvelle rubrique est rattachée à la rubrique Java et visible sur le portail grâce à un bouton "Spring" au même niveau que les rubriques "Eclipse" et "NetBeans", et animée par Hikage ([Lien7](#)), appuyé par les responsables bénévoles Java et les membres de l'équipe bénévole.

Vous retrouverez donc cet espace à l'adresse suivante : <http://spring.developpez.com/> ([Lien8](#))

sans compter :

- Les cours Spring ([Lien9](#))
- La FAQ Spring ([Lien10](#))
- Les critiques de livres Spring ([Lien11](#))
- Les blogs Spring ([Lien12](#))

et bien entendu les forums, à commencer par le forum ci-présent.

Que vous inspire cette initiative ? Avez vous des suggestions / idées pour l'animation de la communauté francophone Spring ?

N'hésitez pas à contribuer à cette nouvelle rubrique en contactant ses responsables (coordonnées en pied de page du portail ([Lien8](#)))

Commentez cette news en ligne : [Lien13](#)

Les livres Java

Ajax pour Java

Comment rendre ses applications web en Java plus dynamiques et réactives ? En mettant de l'Ajax dedans bien sûr ! Grâce aux fonctionnalités typiques d'Ajax, les applications web peuvent s'enrichir de listes de suggestions ou de glisser-déposer de fenêtres qui faciliteront l'accès aux données en temps réel. Ajax pour Java est un guide qui aidera les développeurs Java à se familiariser avec Ajax et les technologies qui s'y rattachent (Struts, Google Web Toolkit, JavaScript, XMLHttpRequest ou encore JSON). Il leur apprendra, par exemple, comment créer des applications s'appuyant sur JavaScript côté-client, lancer des requêtes depuis une servlet Java et en afficher les résultats sans devoir rafraîchir toute la page web, mettre en oeuvre une stratégie client/serveur à base de données brutes avec XML et JSON pour produire des ensembles de données complexes. Il développe entre autres les aspects suivants :

- Le glisser-déposer avec les bibliothèques JavaScript Prototype, script.aculo.us
- La création d'un nuage de tags ; La construction d'un champ de suggestions
- L'intégration d'Ajax dans des applications JSP et JSF
- L'utilisation de Struts dans Ajax avec DWR
- L'exploitation de Google Web Toolkit pour compiler les applications écrites en Java en HTML et JavaScript
- La mise en place de services au client avec GWT

À l'ère du Web 2.0, Ajax pour Java s'adresse aux développeurs Java souhaitant doper leurs applications en

ligne avec Ajax en leur fournissant souplesse, interactivité et réactivité.

Critique du livre par Kamel Kad

Vous ne vous êtes jamais demandé comment une application comme Google Maps est si réactive ? Ou comment certains sites vous suggèrent des informations en fonction de ce que vous êtes en train de saisir dans les champs des formulaires ? Si vous n'avez pas encore de réponse, vous trouverez des réponses dans le livre "Ajax on Java" de Steve Olson, traduit en français par Denis Priou.

Ce livre est destiné aux développeurs souhaitant réaliser des applications web dotées du concept Ajax. Le premier chapitre décrit les étapes nécessaires à l'installation de l'environnement Java, suivi d'un exemple d'interaction entre un client et un serveur via Ajax, dont on trouve la description côté client au chapitre 2 et l'utilisation côté serveur au chapitre 3. Même si on associe souvent XML à Ajax, l'auteur nous montre que cette structure de données n'est pas toujours nécessaire quand il s'agit d'une simple application ou quand il n'y a qu'une donnée à échanger entre le serveur et le client à un instant donné. En revanche si l'application nécessite des transferts de données nombreux et complexes, recourir à une structure comme XML ou JSON est primordial. C'est ce qui est décrit au chapitre 4. Dans ce dernier, l'auteur présente différentes API de parsing XML, notamment JDOM, DOM4J et SAX. Il y décrit aussi la façon de recevoir les données et surtout comment JavaScript exploite le code XML. Dans le même chapitre l'auteur présente une "alternative" à XML. Il s'agit de JSON, un moyen simple et moins gourmand en

ressources, mais qui présente aussi l'avantage de pouvoir effectuer des analyses avec la fonction eval() de JavaScript.

Au chapitre 5, l'auteur décrit les méthodes qui permettent d'atteindre des informations stockées dans une base de données et de les afficher intuitivement grâce à Ajax. Chaque jour qui passe apporte son lot de nouvelles applications web. Pour accélérer les développements, de nombreuses bibliothèques ont vu le jour pour aider les développeurs à créer des sites web utilisant Ajax.

Le chapitre 6 présente ces différentes boîtes à outils, entre autres Dojo Toolkit, Rico Toolkit et DWR (Direct WebRemoting) lesquelles offrent de nombreuses fonctionnalités très utiles. A la fin du chapitre, un exemple de Glisser-Déposer a été détaillé avec la bibliothèque Scriptaculous et Prototype. Cet exemple complet décrit la création d'un panier pour un site e-Commerce.

Si vous êtes amené à réécrire plusieurs fois le même code pour Ajax, sachez que le chapitre 7 vous explique comment développer des fonctionnalités personnelles que vous allez réemployer dans vos applications en créant vos propres étiquettes et utiliser dans les pages JSP.

Le chapitre 8 et 9 présentent les techniques d'intégration d'Ajax dans des applications développées avec les Framework Struts et JSF (Java Server Faces). Le chapitre 10 est un tutoriel présentant l'outil GWT (Google Web Toolkit).

Cet ouvrage est destiné aux développeurs web en Java. Les cours sont présentés d'une manière progressive. Toutefois, les développeurs familiarisés avec les Servlets, JavaScript, HTML et l'analyse syntaxique sous XML n'auront pas de mal à suivre les exemples et les codes sources décrits dans ce livre. Grâce à Ajax vos applications deviendront plus intuitives et réactives. L'auteur l'a dit : "Développeurs Swing, soyez prévenus : Ajax hausse le ton et le niveau de jeu !"

Struts 2 in Action

Critique du livre par Celinio Fernandes

Le livre est divisé en 5 parties.

Dans la 1ère partie les auteurs apportent une présentation sommaire de l'architecture du framework Struts 2 (MVC pattern, interceptors, ValueStack, OGNL) et abordent un exemple Hello World. Les 2 types de configuration sont détaillés: configuration via des fichiers XML (struts.xml) et configuration par des annotations Java.

Dans la 2ème partie les auteurs couvrent l'action dans Struts 2, de façon relativement bien détaillée et bien expliquée. Ensuite un chapitre traite des interceptors : le

principe, la déclaration, les avantages, les interceptors existants et comment créer un interceptor si aucun des interceptors existants ne répond aux besoins. Le langage OGNL, qui permet d'utiliser les objets stockés dans la pile (ValueStack) est abordé pour la première fois dans le livre.

La 3ème partie traite des différents tags. J'ai bien aimé d'abord le découpage rationnel des paragraphes et chapitres suivant les 4 catégories de tags : data tags, control-flow tags, UI tags et autres tags. Ensuite j'ai bien aimé la présentation de chaque tag: d'abord une explication, ensuite un petit tableau décrivant les attributs, types, (...), du tag et enfin un petit extrait de code pour montrer une utilisation concrète. Les auteurs reviennent ensuite sur OGNL et son utilisation pour l'accès aux données provenant de la pile ValueStack (incluse dans ActionContext) mais aussi d'un objet différent d'ActionContext, qui "contient toutes les données qui constituent le contexte d'exécution d'une action". J'ai aussi bien apprécié la présentation rapide d'alternatives aux JSPs telles que Velocity et Freemarker, au niveau des pages de résultats.

La 4ème partie montre comment intégrer les applications Struts 2 avec le plug-in Spring et JPA. Un chapitre entier est consacré au framework bien pratique de validation qui vient avec Struts 2, avec les fichiers XML et les annotations. Le chapitre est lui aussi bien fait, avec abondance de code. Idem pour le chapitre suivant sur l'internationalisation.

Enfin, la 5ème et dernière partie est un peu plus fourre-tout puisqu'elle couvre divers sujets qui, comme le disent les auteurs, nous aident "à travailler plus intelligemment". A savoir l'utilisation de plug-ins (comme le classique Tiles pour la présentation, ou JFreeChart pour la création de graphes), les tests unitaires avec JUnit, comment migrer de Struts 1.X vers Struts 2, comment empêcher le double submit ...

Ce que j'ai aimé dans ce livre : le code source de chaque chapitre est packagé de façon appropriée dans l'archive (fichier war) de l'application finale, sauf les derniers chapitres.

Ce que j'ai moins aimé dans ce livre : J'ai trouvé le livre peut-être un peu trop verbeux à mon goût, dans certains passages. J'aurais préféré que les auteurs aillent d'avantage directement à l'essentiel parfois, surtout quand on dispose de peu de temps pour se former ou que l'on cherche uniquement à acquérir le minimum pour pouvoir développer. L'absence de chapitre au sujet de l'intégration de bibliothèques Javascript et de composants Ajax dans Struts 2.

Le site du livre ([Lien14](#))

Retrouvez ces critiques de livre sur la page livres Java : [Lien15](#)

Developpement Web

Les derniers tutoriels et articles

Maîtriser Firebug, l'indispensable extension Firefox pour le développement web

C'est un fait : le développement web est de plus en plus complexe. Accompagnée de Firefox, l'extension Firebug s'impose comme l'outil incontournable au service du développeur.

Firebug, c'est la possibilité d'inspecter et éditer le code HTML et CSS, de voir, monitorer et déboguer le code JavaScript, de visualiser les échanges entre le serveur et le navigateur... et bien d'autres fonctionnalités encore. Dans cet article, je vous invite à découvrir les principales caractéristiques de Firebug et à en comprendre le fonctionnement.

1. Introduction



La grande utilisation des technologies périphériques à notre bon vieux HTML sonne comme une évidence. Impossible, en effet, de se passer des scripts JavaScript, des feuilles de style (CSS) ou bien des requêtes XMLHTTP (AJAX).

Les navigateurs web, initialement destinés à présenter du contenu statique, sont mis à rude épreuve en prenant en charge certaines tâches "normalement" dévolues aux serveurs.

Conséquence : il devient nécessaire de s'entourer d'outils offrant une aide pour le développement de ces technologies.

Il est possible de trouver différents outils pour le développement des feuilles de style ou des scripts JavaScript, mais ces outils interviennent bien souvent en amont de la visualisation (je pense à certains environnements de développement). Du côté des outils permettant l'intervention "en live" sur les pages, rien de réjouissant non plus.

Firebug répond en grande partie à cette carence en proposant les outils nécessaires pour voir, éditer, déboguer HTML, CSS et JavaScript et en permettant la visualisation instantanée des changements effectués.

Les exemples et les copies d'écran de cet article sont basés sur Firebug 1.3X.3 utilisé avec Firefox 3.0.9 et sous système d'exploitation "Windows Vista Edition Familiale Premium".

L'idée de rédiger un tutoriel sur Firebug est née du manque de ressources francophones traitant du sujet. L'ambition de cet article est de présenter les fonctionnalités qui m'ont parues les plus pertinentes, de permettre une prise en main rapide, enfin de présenter certaines parties moins

accessibles telles que l'utilisation de la console.

1.1. Installation

Firebug est très simple à installer. A partir de la version 3 de Firefox il suffit d'aller dans le menu *outils*, de sélectionner *modules complémentaires* et de choisir l'onglet *catalogue*. Ensuite, saisir "Firebug" et cliquer sur la loupe pour démarrer la recherche.

Attention : la recherche de cette extension retourne également les projets périphériques de Firebug. Il faut donc choisir Firebug. Nous reviendrons en fin d'article sur ces projets périphériques, par ailleurs très intéressants.

Une fois ce module installé, il suffit de redémarrer Firefox et le tour est joué.

1.2. Tour d'ensemble

1.2.1. Les principales fonctionnalités

Il est difficile de donner un bref aperçu des fonctionnalités de Firebug, tant leur nombre est important.

On peut néanmoins distinguer trois axes principaux :

L'inspection et l'édition du code HTML, CSS et du DOM d'une page web.

Le développement des scripts JavaScript : un ensemble d'outils qui permet de voir, déboguer et profiler un script JavaScript.

Le monitoring de l'activité réseau qui permet de visualiser en détail le téléchargement des ressources par le navigateur et, en particulier, d'accéder au détail des requêtes XMLHTTP (AJAX).

Les modifications effectuées n'ont pas d'impact sur les fichiers concernés, ces derniers se trouvant soit sur un serveur, soit sur la machine.

Et l'extension web developer ?

Très complète, l'extension web developer propose un ensemble d'outils permettant un accès simple et rapide aux différents éléments d'une page web : cookies, CSS, formulaires, images...

La valeur ajoutée de Firebug face à cette extension est l'interaction avec le code qui offre de multiples possibilités d'éditer, de déboguer et d'avoir instantanément accès au résultat.

Web developer reste une extension très intéressante et complémentaire de Firebug.

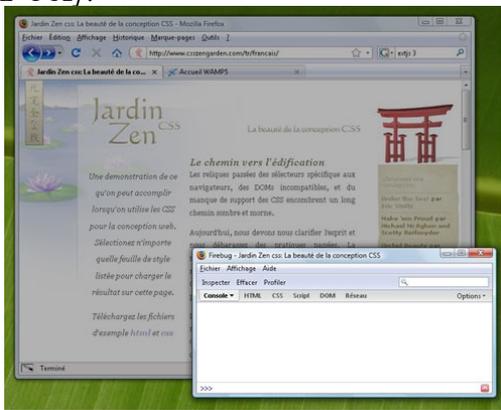
1.2.2. La fenêtre Firebug

Il existe deux manières d'ouvrir la fenêtre Firebug :

Fenêtre intégrée dans le navigateur : en cliquant sur l'icône Firebug située en bas à droite (ou le raccourci clavier [F12]). La fenêtre apparaît en bas de la page comme illustré ci-dessous :



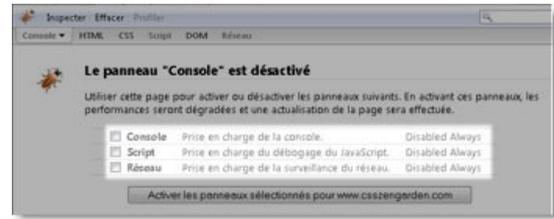
Indépendamment du navigateur : soit en utilisant le clic droit et en choisissant l'élément "Ouvrir Firebug dans une nouvelle fenêtre", soit en utilisant le raccourci clavier [CTRL+F12].



1.3. Prise en main

Une fois l'installation et le redémarrage du navigateur effectués, une petite icône apparaît dans la barre de statut. C'est ici qu'il faut cliquer pour faire apparaître Firebug.

Dans un souci de performance, Firebug se désactive par défaut pour chaque nouveau domaine. La première chose que l'on doit voir est donc un message qui indique que la console est désactivée : "Console panel is disabled"



Cette désactivation concerne :

1. La console
2. Les scripts
3. L'activité Réseau

Cette manipulation est nécessaire **pour chaque nouveau domaine rencontré** au moment où l'on souhaite inspecter une page de ce domaine.

Chaque fichier peut être ouvert dans un éditeur spécifique en passant par le bouton (avec l'icône Firebug) situé sur la gauche de l'écran.

Il est également possible de configurer différents éditeurs dans le gestionnaire d'éditeurs

Firebug peut provoquer le ralentissement de certaines pages.

2. Inspecter et modifier le HTML

2.1. Voir le code source

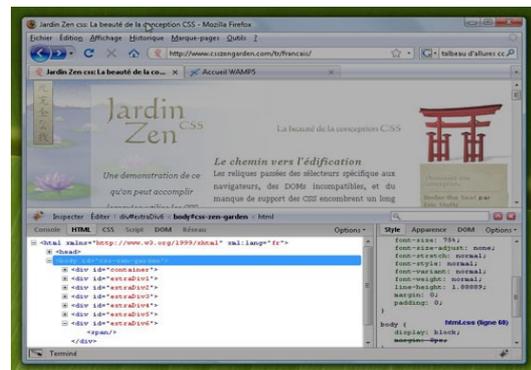
L'onglet HTML permet de visualiser le contenu de la page web affichée dans le navigateur.

Cette vue offre plusieurs avantages en comparaison de la visualisation du code source avec les outils intégrés au navigateur (Firefox ou autre).

D'abord, les différents éléments du code HTML sont colorisés syntaxiquement (bien que la colorisation soit assez basique).

Ensuite le code est automatiquement indenté et présenté à la façon d'un arbre XML où chaque élément est hiérarchisé, la plupart du temps parent ou enfant d'un autre élément.

En voici une petite illustration :

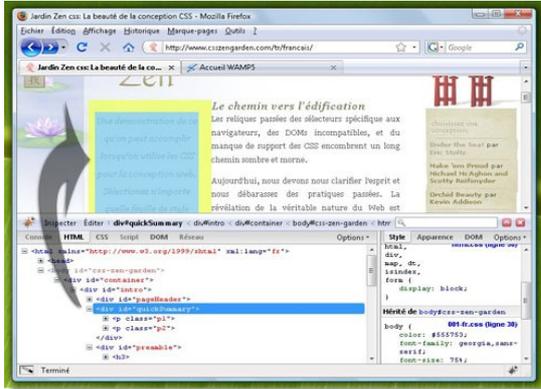


2.2. Retrouver un élément

Firebug propose plusieurs manières de retrouver l'élément HTML d'une page web, en partant du code HTML affiché dans la fenêtre de Firebug, ou bien en partant de l'affichage de la page dans le navigateur.

2.2.1. Inspection dans Firebug

Il est possible de désigner un élément (au survol) dans la fenêtre de Firebug pour qu'il soit mis en surbrillance dans la page affichée par le navigateur.



Cliquer sur cet élément provoque sa sélection dans Firebug, ce qui permet son inspection détaillée (qui sera approfondie un peu plus loin). La sélection d'un élément permet de le retrouver automatiquement après le rechargement de la page.

2.2.2. Inspection dans le navigateur

A l'inverse, voici comment désigner un élément de la page web et le mettre en surbrillance dans Firebug :

Il faut passer en mode inspection et survoler la zone voulue (dans la page web) pour voir le code qui lui correspond dans Firebug.

On peut passer en mode inspection en cliquant sur le bouton *inspecter* ou en utilisant le raccourci : [Ctrl+Shift+C].



Cliquer sur l'élément offre le même mécanisme de sélection que celui évoqué ci-dessus.

2.2.3. La zone de recherche

La zone de recherche permet quant à elle de retrouver une chaîne de caractères. Cette recherche est possible

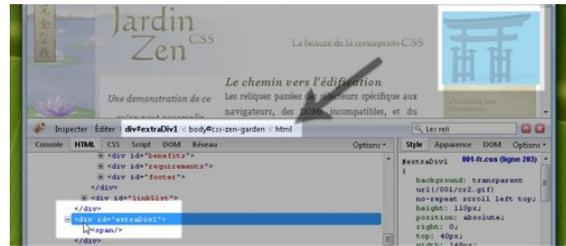
quasiment partout (HTML, CSS, JavaScript et monitoring réseau).



On peut accéder à la barre de recherche soit en cliquant dessus soit par le raccourci [Ctrl+Shift+K].

2.3. Hiérarchie des éléments

Chaque élément sélectionné est situé dans la hiérarchie des éléments HTML à la manière d'un fil d'Ariane, c'est très commode pour situer l'élément dans la hiérarchie.

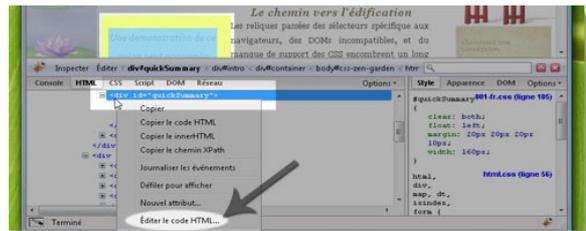


Le survol d'un élément de la hiérarchie permet, comme pour le survol du code, la mise en surbrillance de l'élément correspondant de la page web.

2.4. Edition du code HTML

Au cours de l'inspection du code HTML il est possible d'éditer le code HTML à n'importe quel moment.

Prenons par exemple le cas d'un élément que l'on souhaiterait ajouter dans la page du site de référence : CSS Zend Garden.

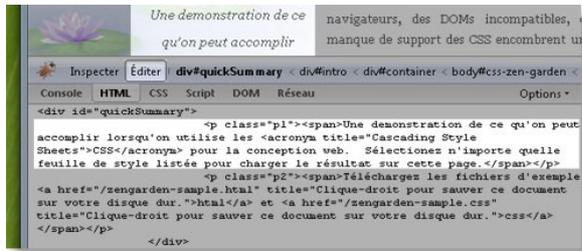


Il existe deux manières de passer en mode édition : soit en cliquant sur le bouton "Edit" situé à droite du bouton d'inspection ou bien en passant par le menu contextuel (clic droit), en choisissant "Edit le code HTML".

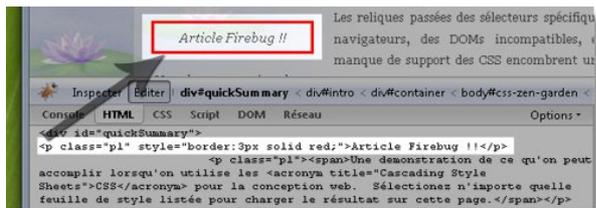
Une fois basculé en mode édition, on retrouve le noeud sélectionné et ses enfants qui apparaissent dans une zone de texte éditable.

A ce stade, tout le code est éditable (texte, HTML, CSS...), les modifications sont prises en compte et visualisables à la volée.

Dans le cadre de notre exemple, nous allons nous intégrer dans le code HTML/CSS existant en créant une nouvelle balise *p* et insérer le contenu "Article Firebug".



Nous avons ajouté au passage quelques propriétés CSS (le bord en rouge par exemple) ce qui permet de distinguer le nouvel élément des autres.



En cliquant de nouveau sur "Editer" (alors que le bouton est sélectionné) nous sortons du mode édition en revenant dans la source HTML de notre page.

On remarque alors que le morceau de code, nouvellement créé, est parfaitement intégré dans le code source.

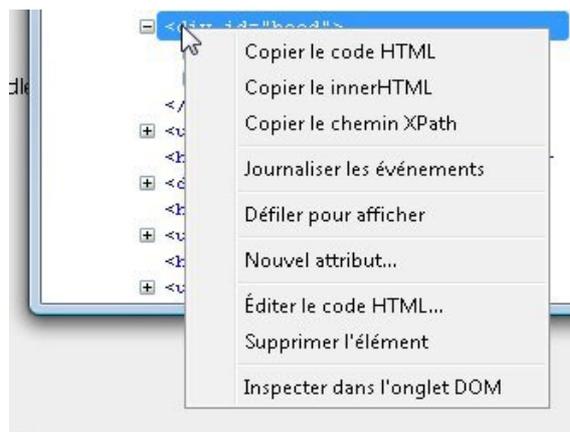


Cet exemple très simple à reproduire illustre la facilité avec laquelle il est possible d'utiliser Firebug pour tester des modifications de code HTML.

2.5. Le menu contextuel

Firebug se repose énormément sur les éléments du menu contextuel pour présenter certaines fonctionnalités qu'on ne trouverait nulle part ailleurs.

Voici les possibilités qu'offre le menu contextuel lors de la sélection d'un élément dans la prévisualisation HTML :



Ces éléments étant suffisamment clairs, je ne détaillerai pas chacun d'entre eux. Je reviendrai simplement sur l'élément de menu "journaliser les événements" qui n'est pas très explicite. Il s'agit simplement d'un monitoring des événements (onclick, onmousemove...) appliqué à l'élément sélectionné. Chaque événement déclenché est alors affiché dans la console.

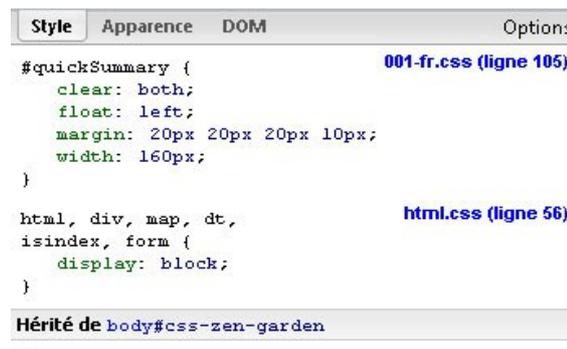
2.6. Plus d'infos : Style, apparence et DOM

Restons dans l'onglet HTML de Firebug. Lorsqu'un élément est sélectionné, la zone située à gauche du code HTML permet d'accéder à trois onglets : *Style*, *Apparence* et *DOM*.



2.6.1. Style

L'onglet *Style* donne accès aux feuilles de style de l'élément sélectionné. Les informations données permettent de connaître la provenance de la règle de style appliquée ainsi que d'éventuelles informations sur son héritage.

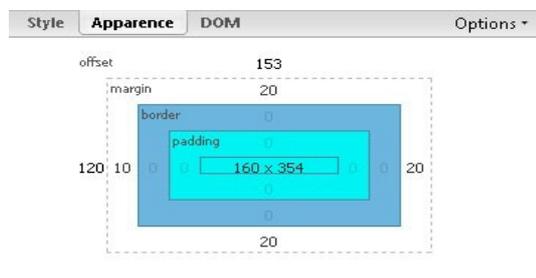


Il est également possible d'éditer une propriété, de la désactiver ou de la supprimer, mais également d'en créer une nouvelle. Cette partie sera abordée en détail dans le chapitre suivant.

2.6.2. Apparence

L'onglet *Apparence* permet la lecture rapide des **mesures d'un élément** sélectionné à savoir :

- La propriété **offset**
- La propriété **margin**
- La propriété **border**
- La propriété **padding**
- La **hauteur** et la **largeur** au centre de cette fenêtre.

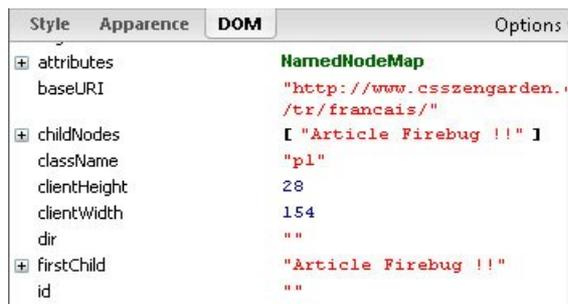


Ces valeurs sont modifiables en cliquant dessus, soit en éditant la valeur au clavier soit en la faisant varier avec les boutons haut et bas. La modification est immédiatement visible selon les mêmes modalités que la modification du code HTML.

2.6.3. DOM (Document Object Model)

Rappelons simplement que le DOM (Document Object Model) est la description d'une interface permettant l'accès et la modification du document (la page web). Communément, c'est en JavaScript que se font les accès au DOM.

Cet onglet donne accès au DOM de chaque élément sélectionné, comme illustré dans la copie d'écran suivante :



3. Des CSS "nickel chrome"

Le développement des feuilles de style en cascade est bien plus complexe qu'il n'y paraît a priori. Pour s'en convaincre, il suffit d'observer les feuilles de style produites par les sites professionnels et d'en constater la complexité.

Firebug propose plusieurs fonctionnalités qui permettent d'y voir plus clair : l'inspection aisée des différents fichiers de CSS, leur visualisation mais aussi la modification de règles de style ou la création de nouvelles propriétés avec la visualisation instantanée du résultat.

L'onglet CSS de Firebug permet un **accès global** aux feuilles de style en cascade (CSS) de la page concernée.

3.1. Choisir le fichier à inspecter

En premier lieu, Firebug permet, via une liste déroulante, de sélectionner le fichier de style à visualiser.

Cette liste comprend :

- Les fichiers définis avec l'instruction CSS "import"
- Les fichiers pointés par l'instruction HTML "link"
- Le style défini dans la page



3.2. Visualiser, éditer

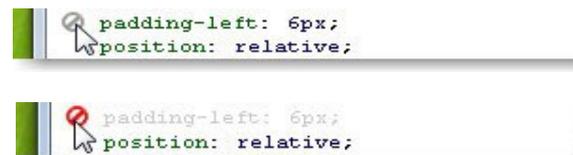
Sélectionner une page permet de visualiser immédiatement la feuille de style dans la zone principale de la fenêtre.



Chaque propriété et sa valeur sont directement éditables, soit en cliquant sur l'élément soit par le menu contextuel (clic droit) en choisissant "Editer". Dans notre exemple, il s'agit du changement de couleur de la police d'un titre.



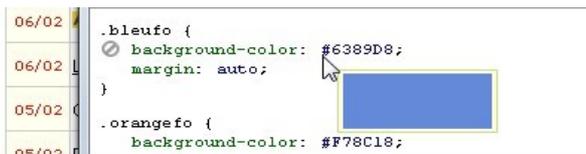
L'exemple suivant illustre la possibilité de **désactiver** une règle de style :



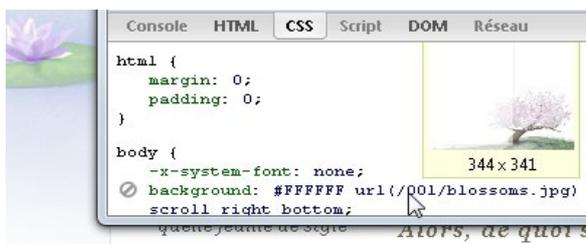
La **création** ou la **suppression** d'une propriété et de sa valeur se font par le menu contextuel.

3.3. Prévisualiser images et couleurs au survol

Les couleurs sont prévisualisables en survolant la propriété qui y fait référence, comme illustré ici :



Un autre exemple pour la prévisualisation d'une image :



Nous avons vu que chaque image pouvait être visualisée en survolant une propriété qui pointe sur une ressource de type image. Il est également possible de **copier le chemin de l'image** ou de **l'afficher dans un nouvel onglet** en passant par le menu contextuel.

3.4. Complétion des propriétés et valeurs possibles d'une instruction CSS

Autre curiosité découverte lors de la rédaction de cet article : Firebug permet la complétion sur les propriétés et les valeurs CSS pendant l'édition ou la création.

Cerise sur le gâteau, les valeurs proposées en complétion sont en accord avec la propriété choisie.

4. Tout pour le JavaScript

Incontestablement, la place du JavaScript dans nos développements est de plus en plus importante.

Aujourd'hui, JavaScript permet au développement web de représenter une alternative crédible aux développements dits « lourds ». Les nombreuses bibliothèques (jQuery, YUI, ExtJS, dojo, etc.) permettent notamment la simulation d'un environnement connecté se reposant en grande partie sur des techniques dites "AJAX".

Ce constat change la donne et pousse à reconsidérer notre manière de développer pour le Web en général et en JavaScript en particulier.

Firebug est un premier pas vers la professionnalisation du développement en JavaScript en proposant **un ensemble d'outils** permettant la **compréhension** du code JavaScript et en rendant la **création** et la **modification** plus aisées.

4.1. Visualiser les erreurs

La première fonctionnalité remarquable de Firebug est l'affichage "amélioré" des erreurs JavaScript.

Firebug affiche les éventuelles erreurs dans la barre de statut de firefox en lieu et place de l'icône Firebug.

En cas d'erreurs, ces dernières sont détaillées dans la console Firebug, comme c'est illustré ci-dessous :



Le signalement d'une erreur est constitué du **message d'erreur**, de la **localisation** de cette erreur dans le code et enfin du **nom de fichier** JavaScript responsable de cette erreur.

Ce nom de fichier est cliquable et amène à la visualisation du fichier concerné dans Firebug.

Firebug affiche également la **pile d'appels des fonctions** dans le contexte de l'erreur.

4.2. Sélection du fichier JavaScript à visualiser

Chaque script JavaScript de la page courante est visualisable dans Firebug (onglet Script). On peut ainsi voir les scripts localisés dans la page et les scripts situés dans des fichiers externes.



On peut filtrer le type de fichier à afficher dans la liste (tous, eval, scripts statiques, scripts événements)

Retrouvez la suite de l'article de Eric Pommereau en ligne : [Lien16](#)

Ce qu'il faut et ne pas faire en CSS : Les balises

Cet article est la traduction de cet article : ([Lien17](#)). Retrouvez toutes les traductions de CSS Globe disponibles sur cssglobe.developpez.com ([Lien18](#)).

CSS Globe poursuit sa série de petits articles appelés "Ce qu'il faut et ne pas faire en CSS". Ces séries ont pour but de pointer les mauvaises habitudes lorsqu'on utilise le CSS et en général les standards du Web. On tentera de répondre aux questions les plus fréquentes relatives au CSS. Cette deuxième partie portera sur les balises.

1. Introduction

Si vous voulez vraiment entrer dans les standards Web, vous découvrirez bientôt que la chose la plus importante est le codage. Un code correct, solide et bien structuré. Le HTML, XHTML, les balises, les attributs, la structure...C'est tout ce dont il s'agit réellement. Nous sommes attachés aux termes des standards Web et CSS. Le terme CSS est presque devenu un synonyme de standard. J'ai souvent utilisé le terme "CSS designer" ou "CSS ci", "CSS ça" pour expliquer rapidement les méthodes et l'approche dont je parlais. Mais la vérité est que le CSS ne signifierait rien si votre structure de codage comporte une erreur. Le codage est le roi.

2. Structurer un document

La clé des documents conformes aux standards Web est leur structure. Elle doit être significative et logique. Le document devrait suivre un certain flux, il doit être lisible et compréhensible sans style CSS.

La plupart des documents ont différentes sections (navigation, contenu principal, contenu latéral, le droit d'auteur, zone de contact, etc.). Un utilisateur, humain ou non humain, devrait être capable de comprendre ces sections comme des sections importantes du premier regard.

La section la plus importante doit être aussi élevée que possible dans le flux, mais l'importance de certaines sections est également définie par la position des balises d'en-tête (h1, h2, h3, h4, h5, h6). En un sens les titres définissent la structure des documents ([Lien19](#)).

3. Balises structurelles et de présentation

Dans le monde des standards Web, on entend souvent parler d'un syntagme "séparant la présentation du contenu". Qu'est-ce que cela signifie ? Fondamentalement, cela signifie que les éléments HTML ne devraient exister que pour la présentation. Vous devriez éviter d'avoir des *div* vident pour les ombres ou autres effets. Laissez aux balises leur sens structurel afin qu'elle existe en tant que partie logique d'un document.

4. Le choix des balises

Il existe une multitude de possibilité dans le choix des balises. Ne comptez pas uniquement sur les *div* pour faire office de conteneur. Parfois, un simple paragraphe, contenant quelques *span* ou *em*, est suffisant pour obtenir

ce que vous voulez. Nous utilisons tous des listes numérotées ou non, mais nous avons aussi un autre type de listes qu'on n'utilise pas assez : les listes de définitions. Les *dl* sont idéales dans de nombreux cas à cause de leur structure à "2 dimensions". D'ailleurs, pourquoi nous n'utilisons pas les balises *address* pour les informations de contact que nous plaçons au bas de la page (dans le "footer") ? Faites votre choix !

5. La validation

La validation a son importance dans l'analyse de votre code. Mais la validation seule ne signifie pas que votre document est construit selon les normes. Vous pouvez avoir l'ensemble du document fait de *div* avec les noms de classes, ce qui permettrait de valider le code, mais serait-ce utile ? Pas à tout le monde et certainement pas aux moteurs de recherche.

6. Résumé

À faire

- Accorder une grande attention à la structure du document. C'est la chose la plus importante ;
- Penser à la sémantique, utiliser les balises dans un ordre significatif, créez un "flux" HTML logique ;
- Utiliser une variété de balises possibles, choisissez la plus appropriée à vos besoins et surtout au type de contenu ;
- Utiliser une variété de balise, choisissez celle appropriée pour un certain type de contenu ;
- Penser à l'aspect utile avant les effets de style ;
- Essayez d'utiliser aussi peu d'éléments HTML que possible pour le même effet, vous vous sentirez plus à l'aise ;
- Faire des compromis en terme de design. Parfois, il est préférable de ne pas utiliser certains effets et d'avoir un code propre.

À ne pas faire

- Penser que si quelque chose est valide il est automatiquement conforme aux standards Web ;
- Sur-exploiter certaines balises, comme les *div* ;
- Utiliser des balises de présentation ;
- Ne pas éviter d'utiliser certains éléments en pensant qu'ils ne sont pas des "standards web" (oui, je veux parler des tableaux) ;
- Faire des compromis en termes de codage.

Retrouvez l'article de Alex Grakalic traduit par Alban Lelasseux en ligne : [Lien20](#)

Les derniers tutoriels et articles

Développement d'un gestionnaire de contacts (CManager) avec ActionScript Foundry (AS Foundry) - Partie 2

Dans ce second tutoriel, nous mettrons en évidence le traitement des réponses du service. Pour réaliser ce tutoriel, il est important d'avoir déjà lu la première partie ([Lien21](#))

1. Gestion de la liste des contacts

1.1. Modification d'un contact

Pour continuer la réalisation du tutoriel nous allons poursuivre à partir du fichier zip téléchargeable ici ([Lien22](#)), celle-ci contient l'ensemble du code source de la partie 1 avec les fichiers nécessaires à la partie 2 que l'on va compléter pendant ce tutoriel.

Pour importer ce projet dans votre workspace, il faudra refaire la même manipulation que dans la partie 1, suivez donc cette procédure ([Lien23](#)).

L'ajout, la modification et la suppression d'un contact utilisent la même procédure que pour la récupération de la liste de contacts, les seules différences sont dans le traitement de la réponse du service. Pour mettre en place ces fonctionnalités, il faut tout d'abord modifier MainView et son helper pour déclencher l'événement de mise à jour d'un contact.

```
//view/MainView.mxml
...
<mx:HBox
    x="800"
    styleName="saveBox"
    id="saveBox"
    moveEffect="{Move}"
    width="250"
    height="40">
    <mx:Label text="Save changes ?"/>
    <mx:Button
        icon="{EmbeddedMedia.getInstance(
).saveContactImg}"
        buttonMode="true"
        styleName="btnTopContact"
        width="30"
        height="30"
        click="{helper.updateContact()}"
    />
...

```

Quand l'utilisateur clique sur le bouton pour faire la mise à jour, la méthode `updateContact` dans `MainViewHelper` est appelée pour lancer le traitement par le contrôleur grâce à la méthode `updateContact` :

```
//helper/MainViewHelper.as
...
/**
 * passe à l'état BTN_SAVE_VIEW_STATE :
```

```
saveBox apparait / topButtonContact disparaît
 * appel à la méthode updateContact dans
MainController
 * */
public function updateContact(): void
{
    getMainView().currentState =
BTN_SAVE_VIEW_STATE;
    if(getMainView().leftSmartForm.is
Update)
    {
        switchMode();
        var updatedContactVO :
ContactVO = getCurrentContact();
        MainController.getInstance().upda
teContact( updatedContactVO );
    }
}
/**
 * Créer un contactVO du contact
actuellement sélectionné
 * @return ContactVO
 * */

public function getCurrentContact() :
ContactVO
{
    var dtLeft : DictionaryTable =
getMainView().leftSmartForm.source;
    var dtRight : DictionaryTable =
getMainView().rightSmartForm.source;
    var dtBottom : DictionaryTable =
getMainView().bottomSmartForm.source;
    dtLeft.putAll( dtRight );
    dtLeft.putAll( dtBottom );
    var contactOut : ContactVO =
ContactVO( MapUtils.getObjectFromDictionaryTable(
dtLeft, ContactVO ) );
    return contactOut;
}
...

```

Le contrôleur appelle le service distant en utilisant la classe `ContactUpdateResponder` :

```
//control/MainController.as
...
/**
 * Appelle updateContact dans businessDelegate et
passe en paramètre le responder qui traite la
réponse
 * */
public function updateContact( pContactVO :
ContactVO ) :void
{
    var updatedContactTO : ContactTO =
```

```
wrapContactVO( pContactVO );
    getMainBusinessDelegate().updateContact(
new ContactUpdateResponder(), updatedContactTO );
}
...

```

Créer la classe `ContactUpdateResponder` qui traite la réponse du service distant et appelle le modèle pour le mettre à jour:

```
//business/responder/UpdateContactResponder.as
...
public class ContactUpdateResponder implements
IBusinessResponder
{
    public function ContactUpdateResponder()
    {
    }
    /**
     * traitement du résultat renvoyé par le
service
     * Mise à jour du modèle
     * */
    public function result(data:Object):void
    {
        var contactVO : ContactVO =
ContactTO( ResultEvent( data ).result ).contacts.
getItemAt( 0 ) as ContactVO;
MainController.getInstance().getContactModel().up
dateThisContact( contactVO );
    }
    public function fault(info:Object):void
    {
        Alert.show("Error updating the
contact !");
    }
}

```

Le modèle met à jour la liste de contacts et notifie les « observers » qui vont rafraîchir l'affichage :

```
//model/ContactModel.as
...
**
* Mise à jour du contact ( en fonction du
contactId )
* Déclenche l'événement indiquant que la liste
de contact à changé
* */
public function updateThisContact( pContactVO :
ContactVO ) : void
{
    var idx : int = 0;
    for each ( var contact : ContactVO in
_contacts){
        if( contact.contactId ===
pContactVO.contactId )
        {
            _contacts.setItemAt( pCon
tactVO, idx );
            _contacts.refresh();
            notifyObservers( new
ContactListNotification( null ) );
            break;
        }
        idx++;
    }
}
...

```

1.2. Ajout d'un contact

L'ajout et la suppression, utilisent exactement le même processus sauf que le traitement est différent dans le modèle. Dans le cas de l'ajout d'un contact :

```
//model/ContactModel.as
...
**
* crée un nouveau contact dans _contacts
* Déclenche l'événement indiquant que la liste
de contact à changé
* */
public function createContact( pContactVO :
ContactVO ) :void
{
    _contacts.addItem( pContactVO );
    _contacts.refresh();
    notifyObservers( new
ContactListNotification( null ) );
}
...

```

1.3. Suppression d'un contact

Lors de la suppression d'un contact, on demande une validation grâce à la classe `Alert` :

```
//helper/MainViewHelper.as
/**
     * Alert demandant confirmation
avant suppression du contact
     * */
    public function
confirmDeleteContact( e : Event ) : void
    {
        Alert.show("Do you really
want to remove "+
selectedContact.firstName.toString() + " " +
selectedContact.lastName.toString() + " from your
contact list ?",
        "Remove " +
selectedContact.firstName.toString() + " " +
selectedContact.lastName.toString() + " :",
        3, null, deleteContact);
    }

```

Dans le cas de la suppression d'un contact :

```
//model/ContactModel.as
...
/**
     * supprime un contact ( en fonction du contactId
)
     * Déclenche l'événement indiquant que la liste
de contact à changé
     * */
public function deleteThisContact( pContactVO :
ContactVO ) :void
{
    var idx : int =
_contacts.getItemIndex( pContactVO );
    var i : int = 0;
    for each (var contact : ContactVO in
_contacts)
    {
        if (contact.contactId ==
pContactVO.contactId)
        {
            _contacts.removeItemAt( i

```

```
);
        _contacts.refresh();
        notifyObservers( new
ContactListNotification( null ) );
    }
    i++;
}
...

```

2. Création du moteur de recherche

Le fonctionnement du « Search Tree » est composé de deux étapes distinctes : premièrement, il faut indexer les données. « Search Tree » crée un graphe d'expressions de recherche et relie les mots aux objets qui les contiennent, d'après les options émises sur les éléments d'objet; deuxièmement, l'arborescence de recherche est à même d'effectuer des recherches en parcourant le graphe d'après une chaîne de caractères qui lui a été soumise. Cette technique est beaucoup plus performante qu'une recherche directe sur la collection, car le plus gros du travail est effectué durant la phase d'indexation. Plus la chaîne de caractères soumise est longue, plus le nombre de possibilités sur le graphe est mince.

2.1. Première étape : construire l'indexation

L'indexation est créée à l'aide des classes SearchTree et SearchProperty. Pour chaque propriété des objets de valeur à indexer, une SearchProperty doit être créée et un type d'indexation doit être défini. Les propriétés sont ensuite utilisées pour créer l'indexation.

```
//helper/MainViewHelper.as
...
/**
 * passe à l'état BASE_VIEW_STATE
 * appelle à la fonction initSearchTree
 * déclare le premier contact de la liste comme
le contact sélectionné
 * */
public function refreshView() : void
{
    baseContactView();
    initSearchTree();
    getMainView().contactList.selectedItem =
getContactModel().getContacts()[0] as ContactVO;
    dispatchEvent( new
Event('selectedContactChange') );
}
...
/**
 * on ContactListNotification :
 * actualise la vue
 * */
override public function update( o : IObservable,
n:Notification) :void
{
    dispatchEvent( new
Event("contactListChange") );
    refreshView();
}
...
/**
 * déclaration du tableau searchProperties :
firstName, lastName
 * début de l'indexation
 * */

```

```
public function initSearchTree() : void
{
    _tree = new SearchTree();
    var searchProperties : Array =
    [
        new SearchProperty( "firstName",
SearchProperty.INDEX_TYPE_CONTAINS ),
        new SearchProperty( "lastName",
SearchProperty.INDEX_TYPE_CONTAINS )
    ];
    _tree.index( getContactModel().getContacts().sour
ce , searchProperties );
}
...

```

Le type d'indexation impactera sur les performances plus que le nombre d'objets à indexer ou à rechercher. Le graphe de mots résultant d'une indexation à l'aide de INDEX_TYPE_CONTAINS sera plus complexes.

2.2. Seconde étape : effectuer la recherche

Pour effectuer la recherche, nous utilisons simplement la méthode de recherche avec une String comme critère de recherche.

```
//helper/MainViewHelper.as
...
/**
 * on application init : renvoie par défaut tous
les contacts sélectionnés dans le modèle
 * Lors d'une recherche : renvoie le résultat de
searchTree dans la liste de contacts du modèle
 * */
[Bindable("contactListChange")]
public function get getContactList() : Array
{
    if( getMainView().filterTextInput.text ==
null || getMainView().filterTextInput.text.length
<1 )
    {
        return
getContactModel().getContacts().source;
    }else{
        var searchResult : Array =
_tree.search( getMainView().filterTextInput.text
).values();
        return searchResult;
    }
}
...

```

On déclenche l'événement getContactList lors de la saisie dans DelayTextInput :

```
//view/MainView.mxml
...
<elements:DelayTextInput
width="100%"
id="filterTextInput"
change="helper.dispatchEvent( new Event('contactListChange') );"
delay="200" />
...

```

En remplissant le champ de recherche en haut à gauche de l'application, la liste de contacts est filtrée.

Retrouvez l'article de Wajdi Hadj ameur en ligne : [Lien24](#)

Une galerie d'images utilisant la propriété CSS z-index et jQuery

Cet article est la traduction de Create a unique Gallery by using z-index and jQuery ([Lien25](#)). Dans cet article, nous allons combiner la propriété CSS z-index et la puissance de jQuery pour réaliser une galerie d'images avec l'apparence d'une pile de photos.

1. Introduction

Dans cet article, nous voulons créer une galerie de photos en utilisant la propriété CSS *z-index*. Dans notre exemple, nous avons une pile de photos. En cliquant sur **Suivant**, nous mettons la première photo en dernière position et à l'inverse, en cliquant sur **Précédent**, la dernière photo se mettra en première position. Tout cela est réalisé uniquement en modifiant la valeur de *z-index* en rajoutant bien sûr une animation pour laisser croire qu'il s'agit bien d'une pile de photos.

Vous pouvez voir un exemple ([Lien26](#)) ou télécharger l'archive ([Lien27](#)).

2. Le XHTML, le CSS et les images

2.1. Le XHTML

```
<!-- début de la partie concernant le tutoriel -->
<div class="grid_6 prefix_1 suffix_1" id="gallery">
  <div id="pictures">
    
    
    
    
    
  </div>

  <div class="grid_3 alpha" id="prev">
    <a href="#previous"><< Image précédente</a>
  </div>
  <div class="grid_3 omega" id="next">
    <a href="#next">Image suivante >>
  </div>
</div>
<!-- fin de la partie concernant le tutoriel -->
```

Nous avons le conteneur global **gallery**, un bloc **pictures** pour les images et deux boutons **prev** et **next** pour changer d'image.

Nous pouvons mettre autant d'images que nous voulons car le script les traite automatiquement.

Si les noms de classes des *div* ne vous disent rien, ils

viennent de 960 Grid System ([Lien28](#)) et ne servent qu'à positionner les blocs. Ils ne sont pas utilisés par le code jQuery et n'interviennent pas dans le fonctionnement de la galerie.

2.2. Le CSS

```
/* début de la partie concernant le tutoriel */
#gallery { position: relative; }
#pictures { position: relative; height: 408px; }
#pictures img { position: absolute; top: 0; left: 0; }

#prev, #next { margin-top: 30px; text-align: center; font-size: 2.0em; }
/* fin de la partie concernant le tutoriel */
```

Le bloc **pictures** est en position *relative* (sous l'en-tête) et a la hauteur d'une image. Les blocs **img** dans le conteneur **pictures** sont en position *absolute*. Comme ils ont tous les propriétés *top* et *left* à 0, ils s'empilent de la dernière image (au-dessus) à la première (en-dessous).



Conteneur des images, position absolue, top et left 0

2.3. Les images

Dans l'exemple, nous utilisons des images au format PNG qui ont toutes un fond transparent et les mêmes dimensions. Pour créer l'effet de pile, nous avons ajouté une légère ombre portée et une petite rotation dans différentes directions. Je l'ai fait avec Photoshop, mais si vous voulez automatiser cela, vous pouvez utiliser Image Magick ([Lien29](#)) ou regarder du côté des canvas ([Lien30](#)).

3. Le principe de changement de z-index

La propriété *z-index* va représenter la position de l'image, 1 correspond à la dernière position et 5 (il n'y a que 5 images dans l'exemple) à la première.

En cliquant sur **Image suivante**, nous voulons mettre la

première image en dernière position, c'est-à-dire affecter son *z-index* à 1. Tous les autres *z-index* devront être incrémentés de 1. Ainsi, l'image qui était en deuxième position (4) se retrouvera en première place et ainsi de suite.

En cliquant sur **Image précédente**, nous voulons mettre la dernière image en première position, c'est-à-dire affecter son *z-index* à 5. Tous les autres *z-index* devront être décrémentés de 1. Ainsi, l'image qui était en première position (5) se retrouvera en deuxième place et ainsi de suite.

4. Le code

```
$(document).ready(function() { //attend que le
DOM soit prêt pour commencer
    var z = 0; //initialise les z-index
    var inAnimation = false; //flag pour tester si
nous sommes en cours d'animation

    $('#pictures img').each(function() { //attribue
les z-index de départ
        z++; //à la fin, le plus haut z-index sera
dans cette variable
        $(this).css('z-index', z); //attribue le z-
index à la balise <img>
    });

    function swapFirstLast(isFirst) {
        if(inAnimation) return false; //si une
animation est en cours, on ne fait rien
        else inAnimation = true; //modifie le flag
pour dire que l'animation est en cours

        var processZindex, direction, newZindex,
inDeCrease; //prévoit le cas précédent ou suivant

        if(isFirst) { processZindex = z; direction =
'-'; newZindex = 1; inDeCrease = 1; } //attribue
les variables dans le cas 'suivant'
        else { processZindex = 1; direction = '';
newZindex = z; inDeCrease = -1; } //attribue les
variables dans le cas 'précédent'

        $('#pictures img').each(function() {
//actualise chaque image
            if($(this).css('z-index') == processZindex)
{ //s'il s'agit de l'image à animer
                $(this).animate({ 'top' : direction + $
(this).height() + 'px' }, 'slow', function() { //
anime l'image au-dessus/en-dessous (les images
doivent avoir la même hauteur)
                    $(this).css('z-index', newZindex)
//attribue le nouveau z-index
                    .animate({ 'top' : '0' }, 'slow',
function() { //ramène l'image à sa position
initiale
                        inAnimation = false; //modifie le
flag pour dire que l'animation est finie
                    });
                });
            } else { //ce n'est pas l'image à animer,
on modifie juste le z-index
                $(this).animate({ 'top' : '0' }, 'slow',
function() { //il faut attendre la fin de
l'animation pour modifier le z-index
                    $(this).css('z-index', parseInt($
(this).css('z-index')) + inDeCrease); //ajuste le
z-index
                });
            }
        });
    });
});
```

```
        return false; //on ne suit pas le href du
lien
    }

    $('#next a').click(function() {
        return swapFirstLast(true); //met la première
image en dernière position
    });

    $('#prev a').click(function() {
        return swapFirstLast(false); //met la
dernière image en première position
    });
});
```

Regardons de plus près les points importants.

```
$('#pictures img').each(function() { //attribue
les z-index de départ
    z++; //à la fin, le plus haut z-index sera
dans cette variable
    $(this).css('z-index', z); //attribue le z-
index à la balise <img>
});
```

Nous n'avons pas fixé de *z-index* dans les balises ni en CSS. Pour avoir un point de départ, nous l'attribuons avec jQuery. L'ordre est celui d'apparition des balises dans le code. A la fin de la boucle sur toutes les images du conteneur **pictures**, nous obtenons le nombre d'images et la valeur maximale de *z-index* dans la variable **z**

```
function swapFirstLast(isFirst) {
```

Nous utilisons la même fonction pour les deux animations.

```
if(inAnimation) return false; //si une animation
est en cours, on ne fait rien
else inAnimation = true; //modifie le flag pour
dire que l'animation est en cours
```

Nous ne voulons qu'une animation à la fois. Le *flag inAnimation* nous permet de savoir si une animation est en cours. Si c'est le cas, nous sortons directement de la fonction, sinon, on en lance une.

```
var processZindex, direction, newZindex,
inDeCrease; //prévoit le cas précédent ou suivant

if(isFirst) { processZindex = z; direction = '-';
newZindex = 1; inDeCrease = 1; } //attribue les
variables dans le cas 'suivant'
else { processZindex = 1; direction = '';
newZindex = z; inDeCrease = -1; } //attribue les
variables dans le cas 'précédent'
```

Comme nous n'utilisons qu'une fonction pour deux actions différentes, nous devons ajuster les variables. **processIndex** est la position que nous devons animer, soit la première (suivante) soit la dernière (précédente). **direction** correspond à où l'image va devoir aller, soit au-dessus, soit en-dessous de la pile. **newZindex** sera attribué à l'image à animer, 1 pour la dernière position ou la valeur de **z** pour la première. Enfin, mais pas des moindres, **inDeCrease** permet de savoir si les *z-index* seront incrémentés ou décrémentés.

```
$('#pictures img').each(function() { //actualise
chaque image
```

On boucle sur les images pour modifier les *z-index*.

```
if($(this).css('z-index') == processZindex) {
//s'il s'agit de l'image à animer
$(this).animate({ 'top' : direction + $
(this).height() + 'px' }, 'slow', function() { //
anime l'image au-dessus/en-dessous (les images
doivent avoir la même hauteur)
$(this).css('z-index', newZindex) //attribue le
nouveau z-index
.animate({ 'top' : '0' }, 'slow', function()
{ //ramène l'image à sa position initiale
inAnimation = false; //modifie le flag pour
dire que l'animation est finie
});
});
};
```

D'abord, on vérifie s'il s'agit de l'image à animer. Si c'est le cas, on lance la première animation. Lorsqu'elle est terminée, on attribue le nouveau *z-index* puis on lance la seconde animation. Une fois l'image revenue à sa place, on modifie le *flag inAnimation*.

```
else { //ce n'est pas l'image à animer, on
modifie juste le z-index
$(this).animate({ 'top' : '0' }, 'slow',
function() { //il faut attendre la fin de
l'animation pour modifier le z-index
$(this).css('z-index', parseInt($
(this).css('z-index')) + inDeCrease); //ajuste le
z-index
});
}
```

Si l'image n'est pas celle à animer nous avons juste à modifier le *z-index*. Nous récupérons son *z-index* et l'incrémentons / décrémentons selon le cas. Nous lançons une animation (qui ne fait rien) pour être sûrs de ne mettre à jour le *z-index* qu'une fois que la première animation (placer l'image au-dessus des autres) est terminée.

```
$('#next a').click(function() {
return swapFirstLast(true); //met la
première image en dernière position
});
$('#prev a').click(function() {
return swapFirstLast(false); //met la
dernière image en première position
});
```

Ici, nous affectons au clic sur les boutons **prev** et **next** l'appel de la fonction. Nous retournons le résultat de la fonction (qui vaut toujours *false*) de façon à éviter de suivre le lien.

5. Conclusion

Nous avons créé une galerie de bonne allure avec juste quelques lignes de code. Bien sûr, on pourrait ajouter d'autres fonctionnalités (image aléatoire, mélange des images...) mais maintenant, vous savez comment combiner des propriétés CSS et la puissance de jQuery. N'hésitez pas à modifier le code selon vos besoins et de nous le faire partager. Vous pouvez aussi laisser vos commentaires.

Retrouvez l'article de Sebastian Senf traduit par Didier Mouronval en ligne : [Lien31](#)

Améliorer la galerie z-index avec un preloader

Cet article est la traduction de l'article Enhancing the z-index Gallery with a preloader ([Lien32](#)). Dans cet article, nous allons améliorer notre galerie d'images détaillée dans l'article précédent. Nous allons ajouter un indicateur de préchargement d'images pour obtenir une meilleure apparence de la galerie. Tout cela en même pas 10 lignes de code !

1. Introduction

Dans notre précédent article ([Lien31](#)), nous avons créé une galerie d'images plutôt sympa. Mais elle est incomplète pour un site en production. Au premier chargement, vous avez pu constater que les images se chargent et s'affichent très lentement. Pour avoir une apparence plus agréable, nous allons précharger toutes les images et afficher un indicateur de préchargement pour montrer aux utilisateurs que les images sont en chargement. Nous allons faire ça avec à peine 10 lignes de code ! Vous pouvez voir un exemple ([Lien33](#)) ou télécharger l'archive originale ([Lien34](#)).

2. Le CSS pour le calque de chargement.

Donc, nous voulons indiquer que la galerie est en cours de chargement. Nous allons récupérer une animation de chargement sur [ajaxload.info](#) ([Lien35](#)). Ensuite, nous aurons besoin d'une *div* par-dessus les images et les boutons.



Le calque de préchargement

Nous allons appliquer les styles avec du CSS et nous ajouterons la balise avec jQuery lorsque le DOM sera chargé. Comme cela, nous n'aurons pas à toucher au code XHTML (ce qui est pratique si vous avez plusieurs galeries).

Le CSS pour le calque ressemble beaucoup à celui des images. Il est positionné en *absolute* et aligné sur le haut gauche du conteneur *pictures* :

```
#loader {
    position: absolute;
    top: 0;
    left: 0;
    height: 458px;
    width: 100%;
    background: url(../images/ajax-loader.gif) white no-repeat center center;
    z-index: 9999;
}
```

La première différence est la propriété *height*, en effet, nous voulons masquer aussi les boutons. Nous ajoutons donc la hauteur des images et celle des boutons. Comme le calque est en position *absolute*, il se placera par-dessus le conteneur des boutons aussi. Ensuite, nous avons la petite animation centrée en *background*. Enfin, avec un grand *z-index*, nous sommes sûrs qu'il sera au-dessus de toutes les images.

3. Insérer le calque dans le DOM et précharger les images.

Ajouter le calque est très facile grâce à jQuery :

```
$('#pictures').append('<div id="loader"></div>');
//ajoute la div loader, par-dessus les images
```

La *div* sera ajoutée dès que le DOM sera prêt, nous voulons que le calque disparaisse progressivement lorsque les images seront chargées. Voici le code pour pouvoir faire cela :

```
$('#pictures img').each(function() { //attribue
    les z-index de départ
    z++; //à la fin, le plus haut z-index sera dans
    cette variable
    $(this).css('z-index', z); //attribue le z-
    index à la balise <img>

    $(new Image()).attr('src', $
    (this).attr('src')).load(function() { //crée un
    nouvel objet image et appelle une fonction quand
```

```
elle est chargée
    imgLoaded++; //une image de plus est chargée

    if(imgLoaded == z) { //toutes les images
    sont-elles chargées ?
        $('#loader').fadeOut('slow'); //si c'est le
    cas, le calque disparaît
    }
    });
});
```

Nous avons simplement étoffé la boucle d'attribution des *z-index*. Détaillons un peu cette nouvelle portion de code.

```
$(new Image()).attr('src', $
(this).attr('src')).load(function() { //crée un
nouvel objet image et appelle une fonction quand
elle est chargée
```

Cette ligne est particulièrement magique ! Nous créons une nouvelle image avec l'attribut *src* correspondant à l'image que nous voulons précharger. Nous lui affectons l'événement **load** pour déterminer quand elle sera chargée.

```
imgLoaded++; //une image de plus est chargée
```

Si une image est chargée nous incrémentons un compteur.

```
if(imgLoaded == z) { //toutes les images sont-
elles chargées ?
    $('#loader').fadeOut('slow'); //si c'est
le cas, le calque disparaît
}
```

Ensuite, nous vérifions si le nombre d'images chargées et égal au nombre total d'images. Si c'est le cas, nous faisons disparaître le calque.

4. Conclusion

Encore une fois, nous voyons comment quelques lignes de code jQuery (3 dans notre cas) peuvent permettre de régler certains problèmes. Maintenant, notre galerie est complète et peut être utilisée sur notre site.

Retrouvez l'article de Sebastian Senf traduit par Didier Mouronval en ligne : [Lien36](#)

Ruby on Rails

Les derniers tutoriels et articles



Commencer avec Ruby on Rails

Dans cet article, découvrez les différents composants de Ruby on Rails et prenez en main le framework. Découvrez les atouts de ce framework qui ne laisse pas indifférent.

1. Introduction

Ce guide s'inspire du site « Rails Guides » ([Lien37](#)), de mon expérience personnelle de Ruby on Rails et dans tout un tas d'autres ressources (screencasts, sites communautaires, etc...).

Ce guide est également l'occasion de constituer une application de A à Z, d'en voir tous les tenants et les aboutissants (conception, codage, déploiement, etc...). Je vous souhaite d'ores et déjà une très bonne lecture !

2. Recommandations

Dans ce guide, je pars du principe que le lecteur possède une connaissance minimale du langage Ruby. Si certaines portions de code Ruby nécessitent des précisions, j'expliquerai le plus clairement possible ces points de détail.

Pour commencer avec Ruby on Rails, il est nécessaire d'avoir au moins trois choses installées :

- Ruby ([Lien38](#)) (version 1.8.7 dans le cas de ce guide)
- Rubygems ([Lien39](#)) (version 1.3.2 dans le cas de ce guide)
- SQLite ([Lien40](#)), MySQL ([Lien41](#)), ou PostgreSQL ([Lien42](#))

3. Qu'est ce que Rails ?

Rails est un framework de développement web écrit en langage Ruby. Il est conçu pour rendre le développement d'applications web le plus simple et le plus concis possible. Tous les développeurs qui se sont mis un jour à Rails s'accordent à dire que le développement avec Rails est plus fun. Ceci est lié en grande partie au fait que le langage Ruby soit si intuitif.

Le développement avec Rails nécessite l'apprentissage de bonnes pratiques de développement. Une fois ces bonnes pratiques appliquées, le développeur gagne en productivité. Si le développeur garde des réflexes de développement acquis sur d'autres langages, ou utilise des design patterns vus dans d'autres contextes, il aura une moins bonne expérience de Rails et n'en tirera pas tous les bénéfices.

La philosophie de Rails se base sur trois principes :

- Le DRY (Don't Repeat Yourself) : Rails encourage à ne pas écrire plusieurs fois le même code. Rendre réutilisable le code de vos applications est une bonne chose !
- Convention over configuration : Rails se base

uniquement sur des conventions, non sur de la configuration (à l'inverse d'autres langages comme Java par exemple). Si vous respectez ces principes de convention, vous rendrez votre code maintenable et vous gagnerez en productivité.

- REST : c'est le meilleur design pattern pour le développement d'applications web. Il vous permet d'organiser vos applications autour de verbes HTTP standards.

3.1. L'architecture MVC

Rails est organisé selon l'architecture MVC (Modèle-Vue-Contrôleur). Cela lui permet :

- D'isoler la logique métier de l'interface utilisateur
- De maintenir le code « DRY »
- De conserver le code là où il doit résider, pour une meilleure maintenance

3.2. Les différents composants de Rails

Action Controller

C'est le composant qui gère les contrôleurs de l'application. Les services rendus par le composant sont la gestion de session, le rendu de templates, et la gestion de redirection.

Action View

Gère la partie vue de vos applications. Il permet de créer du HTML et du XML par défaut. Les services rendus par ce composant sont le rendu de templates, l'inclusion et le rendu de templates partiels (« partials »), et le support d'AJAX.

Active Record

C'est la base des modèles dans vos applications Rails. Il fournit l'indépendance vis-à-vis des bases de données, la base des fonctionnalités CRUD (Create-Retrieve-Update-Delete), et la capacité à relier les modèles entre eux par exemple.

Active ressource

Ce composant fournit un framework pour gérer la connexion entre des objets métier et des services web RESTful. Il implémente une façon de mapper des ressources web à des objets locaux avec une sémantique CRUD.

Active Support

C'est une collection extensive de classes utilitaires et d'extensions Ruby standards, utilisées à la fois dans vos applications et dans le code du framework Rails.

3.3. REST

Les principes de base de REST sont les suivants :

- L'utilisation d'URLs pour représenter des ressources.
- L'utilisation de verbes HTTP pour toutes les opérations nécessaires (GET, POST, PUT et DELETE)
- L'utilisation des standards HTML et XML

Par exemple, dans une application Rails une requête du type :

```
DELETE /photos/17
```

fait référence à une ressource de type photo, d'identifiant 17, et l'action entreprise est de type suppression.

Si vous souhaitez plus de précisions sur REST, il est conseillé de consulter l'article "A brief introduction to REST" ([Lien43](#))

4. Création d'un projet Rails

4.1. Installation de Rails

La première opération à effectuer avant de créer un projet Rails est évidemment d'installer Rails. Rails est disponible sous la forme d'un gem ([Lien44](#)). Pour l'installer, lancer la commande :

```
gem install rails
```

La version de Rails utilisée lors de ce tutoriel est la version 2.3.2. Elle sert de base pour tout le reste du présent document.

4.2. Création de l'application « Mes Contacts »

Dans le cadre de ce tutoriel, nous allons constituer une application complète de gestion des contacts. Cette application devra fournir :

- Une gestion multi-utilisateurs
- Une gestion personnelle de contacts avec leurs caractéristiques
- Un moyen d'importer des contacts provenant d'autres applications (gmail, hotmail, ...)
- Un moyen d'exporter des contacts sous plusieurs formes
- Un code source de haute qualité répondant aux standards Rails

Procédons à la création du projet. Tout ce que vous avez à faire est de taper :

```
rails mycontacts
```

Lorsque vous lancez cette commande, la base de données qui va être utilisée par défaut est SQLite. Si vous souhaitez utiliser une autre base de données, il suffit de suffixer la commande avec l'option -d et le nom de la base de données. Exemple pour MySQL :

```
rails mycontacts -d mysql
```

Positionnez-vous maintenant dans le dossier de l'application avec la commande :

```
cd mycontacts
```

Si vous listez le contenu du répertoire, vous obtiendrez la structure standard d'un projet Rails. En voici le détail :

Fichier / Répertoire	Description
app/	Contient le code Rails de l'application (modèles, vues, contrôleurs...)
config/	Contient le code de configuration de l'application
db/	Contient le code de description de la base de données
doc/	Contient la documentation du code de l'application
lib/	Contient les modules d'extensions éventuels de l'application
log/	Contient les traces de l'application
public/	Contient les fichiers statiques de l'application (images, javascript, css)
script/	Contient les scripts Rails nécessaires au développement
test/	Contient les tests unitaires de l'application
tmp/	Contient les fichiers temporaires générés par l'application
vendor/	Contient les gems et plugins nécessaires à l'application

4.3. Configuration de la base de données

Notre application de gestion des contacts va reposer sur une base de données, ils nous faut donc la configurer. Lorsque nous avons créé le projet tout à l'heure, Rails a créé pour nous un fichier de configuration de la base de données, qui se situe dans config/database.yml. Ce fichier est un fichier de configuration au format [YAML](#). YAML est le format de fichier de configuration par défaut utilisé dans tous les projets Rails. C'est un format assez simple à appréhender.

Sachez tout d'abord que tout projet Rails est créé avec trois environnements de travail par défaut :

- **development** : lorsque vous développez votre application, c'est dans ce mode que vous travaillez, c'est également celui par défaut. Chaque modification que vous ferez sur votre code dans ce mode sera prise en compte « à chaud ».
- **test** : c'est le mode utilisé lorsque vous souhaitez lancer vos tests unitaires d'application.
- **production** : c'est le mode utilisé lorsque votre projet est finalisé. Il contient des optimisations significatives qui font la différence lors de l'exécution.

Voyons maintenant à quoi ressemble notre fichier de configuration database.yml, en ne prenant en considération que la partie de l'environnement « development » :

```
config/database.yml
# SQLite version 3.x
# gem install sqlite3-ruby (not necessary on
OS X Leopard)
development:
  adapter: sqlite3
  database: db/development.sqlite3
  pool: 5
  timeout: 5000
```

Commençons par le commentaire : on nous conseille d'installer le gem sqlite3-ruby, pour avoir le support SQLite. Faisons-le :

```
gem install sqlite3-ruby
```

Le paramètre « adapter » décrit le type de base de données. Il peut prendre la valeur du nom d'une quantité impressionnante de bases de données, toutes supportées par Rails. En voici une liste non exhaustive : SQLite, MySQL, PostgreSQL, Oracle, DB2, ...

Le paramètre « database » contient en général le nom de la base de données. Dans le cas de SQLite, il contient le chemin vers le fichier qui contient la base de données. Ce fichier n'existe pas encore, mais nous verrons plus loin comment le créer.

Le paramètre « pool » permet de déterminer le nombre de connexions à allouer pour la base de données.

Le paramètre « timeout », quant à lui, permet de déterminer, en millisecondes, le temps maximum à allouer à une requête sur la base de données.

4.4. Création de la base de données

Maintenant que la base de données est configurée, nous devons la créer. Pour cela, il suffit de taper :

```
rake db:create
```

Rake est une commande d'usage général pour les projets Rails entre autres. Pour obtenir une liste des commandes Rake disponibles, tapez « rake -T ».

5. Bonjour, Rails !

Un des moyens d'apprendre un nouveau langage est de créer un programme court pour afficher quelque chose à l'écran. Pour effectuer cela avec Rails, vous devez créer au moins un contrôleur et une vue.

Tapez la commande suivante :

```
ruby script/generate controller home index
```

Rails crée ici plusieurs fichiers pour vous, dont le fichier contrôleur app/controllers/home_controller.rb, et le fichier vue app/views/home/index.html.erb. Ouvrez le fichier index.html.erb, et complétez-le avec ce qui suit :

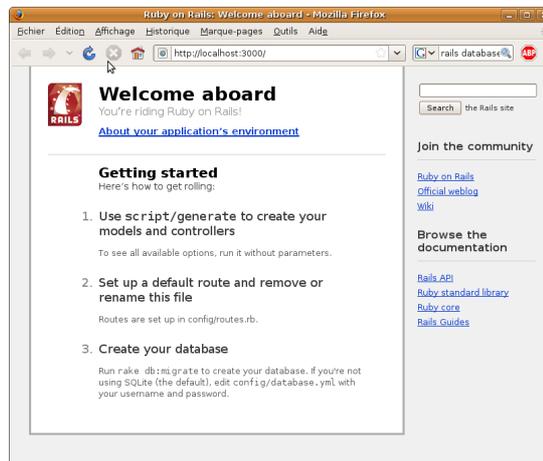
```
app/views/home/index.html.erb
<h1>Bonjour, Rails!</h1>
```

5.1. Démarrage du serveur

Vous avez une application Rails fonctionnelle après avoir exécuté deux commandes seulement. Pas mal du tout ! Démarrez maintenant le serveur d'application Rails en lançant la commande :

```
ruby script/server
```

Le serveur démarre par défaut sur le port 3000. Ouvrez un navigateur internet à l'adresse : http://localhost:3000. Voici ce que vous devriez voir :



Pour voir la page que vous venez de créer, ouvrez le navigateur à l'adresse : http://localhost:3000/home/index.

5.2. Positionnement de la page par défaut

Vous voudrez probablement remplacer la page d'accueil par défaut de Rails par la page que vous venez de créer. Pour cela, vous devez supprimer le fichier index.html qui se situe dans le répertoire public :

```
rm public/index.html
```

Ensuite, vous devez configurer ce qui s'appelle les « routes », c'est à dire les chemins qui permettent de faire le lien entre des URL et les contrôleurs de l'application. Cela se passe dans le fichier config/routes.rb. À la fin du fichier, vous trouverez la configuration des chemins par défaut :

```
config/routes.rb
map.connect ':controller/:action/:id'
map.connect ':controller/:action/:id.:format'
```

Cela signifie que le format des URLs est du style : contrôleur/action/identifiant. Par exemple, si nous voulions dans notre application de gestion des contacts supprimer un contact d'identifiant « 17 » (à compter que cette fonctionnalité soit implémentée), nous pourrions le faire via l'URL : http://localhost:3000/contacts/delete/17.

Revenons au cas de notre page par défaut. Il faut que le chemin par défaut de l'application pointe sur notre contrôleur, et plus précisément sur l'action « index ». Rajoutez ce qui suit au fichier routes.rb :

```
map.root :controller => "home"
```

Cela illustre le principe « convention over configuration » de Rails : nous n'avons pas précisé le nom de l'action dans la ligne ci-dessus, et pourtant la page d'accueil pointe bien maintenant sur home/index. Interrompez le serveur avec CTRL+C, relancez-le, et constatez le résultat en appelant `http://localhost:3000`.

6. Le système de « scaffolding »

C'est une des fonctionnalités puissantes et un peu « magiques » de Rails. Elle permet à partir de très peu de commandes et de code d'avoir un CRUD (Create-Read-Update-Delete, les 4 fonctions basiques d'une base de données) ([Lien45](#)) complet sur un composant donné.

6.1. Création des contacts

Nous allons créer maintenant la partie données représentant les contacts de notre application. Nous ne serons pas exhaustifs, car cela fera l'objet de futurs tutoriels. Pour créer l'ensemble des fichiers nécessaires au CRUD, tapez la commande suivante :

```
ruby script/generate scaffold Contact name:string
email:string
```

Nous ne verrons pas ici le détail des fichiers générés. Nous pouvons juste voir que nous avons créé une entité « Contact » avec comme attributs un nom et un email. Avant de pouvoir profiter du CRUD, nous devons d'abord créer la table correspondant à cette entité « Contact » en base de données. Pour cela, tapons la commande :

```
rake db:migrate
```

Ensuite, mettons à jour le fichier `routes.rb` pour qu'il pointe sur le contrôleur correspondant :

```
config/routes.rb
map.root :controller => "contacts"
```

Voici maintenant la nouvelle page d'accueil de l'application :



Cliquez sur « New contact » pour ajouter des contacts dans la liste, les modifier, les supprimer, ou encore les lister. Vous avez ici une première ébauche de l'application. Certes, ce n'est pas très personnalisé, mais vous n'avez pratiquement écrit aucune ligne de code.

6.2. La console Rails

La console Rails est une invite de commandes irb qui met

à notre disposition toutes les classes de l'application Rails en cours. Vous pouvez donc par exemple interagir sur la base de données à volonté. Pour la lancer, il faut taper :

```
ruby script/console
```

Ensuite, vous pouvez par exemple lister les contacts existants :

```
>> Contact.find(:all)

=> [#<Contact id= "1" name= "Vincent Dubois"
email= "duboisv@hotmail.com" created_at= "2009-
04-18 17:26:55" updated_at= "2009-04-18
17:26:55">,
  #<Contact id= "2" name= "Albert Dupont" email=
"adupont@mycontacts.fr" created_at= "2009-04-19
08:05:02" updated_at= "2009-04-19 08:05:02">]
```

Nous pouvons aisément constater que l'appel à `find(:all)` a ramené le jeu complet des données de la table « contacts », dans un tableau. Vous pouvez si vous le souhaitez créer des contacts :

```
>> my_contact = Contact.new(:name => "Pierre
Martin", :email => "pm@mycontacts.fr")
=> #<Contact id= nil name= "Pierre Martin"
email= "pm@mycontacts.fr" created_at= nil,
updated_at= nil>

>> my_contact.save
=> true
```

Ou même en supprimer :

```
>> my_contact.destroy

=> #<Contact id= 3, name= "Pierre Martin",
email= "pm@mycontacts.fr", created_at= "2009-04-
19 10:02:54", updated_at= "2009-04-19 10:02:54">
```

Pour sortir de la console Rails, faites-le comme dans irb, avec la commande « exit ».

Voilà, vous avez les premières clés en main pour vous lancer dans l'exploration des répertoires d'une application Rails. N'hésitez pas à vous balader, aller voir le contenu des fichiers contrôleurs, des fichiers modèles, ou encore des fichiers vues, modifiez-les, observez les résultats. Bref, faites connaissance avec les possibilités de l'API Rails. Vous trouverez toute la documentation concernant cette API à l'adresse suivante : ([Lien46](#)).

Dans le prochain chapitre, nous entrerons en profondeur dans le sujet des bases de données. Nous étudierons le composant Active Record, et plus particulièrement les migrations. Nous en profiterons pour peaufiner notre base de données.

Vous pouvez retrouver le source de ce tutoriel ([Lien47](#)).

Retrouvez l'article de Vincent Dubois en ligne : [Lien48](#)

Créer une arborescence de fichiers avec jQuery

Un plugin AJAX d'exploration de fichiers pour jQuery.

1. Présentation

L'explorateur de fichiers jQuery est un plugin AJAX, configurable pour jQuery ([Lien49](#)). Vous pouvez créer une arborescence personnalisée de fichiers, entièrement interactive avec une simple ligne de code JavaScript.

Actuellement, côté serveur, les scripts de connexion sont disponibles pour PHP, ASP, ASP.NET, JSP et Lasso. Si vous êtes développeur, vous pouvez facilement créer votre propre script de connexion pour travailler avec le langage de votre choix.

2. Fonctionnalités

- Produit du XHTML valide et sémantique ;
- Entièrement personnalisable via CSS ;
- Possibilité de changer l'icône basée sur l'extension de fichier ;
- Utilisation d'AJAX pour aller chercher des informations sur les fichiers à la volée ;
- Facile à configurer et à mettre en oeuvre ;
- Contient les scripts de connexion pour PHP, ASP, ASP.NET (C#), JSP et Lasso ;
- Possibilité de prise en charge de scripts de connexion personnalisés ;
- Personnalisation possible des événements «Développeur/Réduire» ;
- Personnalisation possible de la vitesse des événements «Développeur/Réduire» ;
- Prise en charge des fonctionnalités jQuery 'easings' (effets de jQuery) ;
- Vue multi-dossier ou dossier unique ;
- Message de chargement personnalisable.

3. Compatibilité

L'explorateur de fichiers jQuery fonctionne sur tous les navigateurs supportés par jQuery. Il a été entièrement testé sur :

- Internet Explorer 6 & 7
- Firefox 2 & 3
- Safari 3
- Chrome (bêta)
- Opera 9

4. Démo

Voir une démonstration ([Lien50](#)) de l'explorateur de fichiers avec diverses options.

5. Téléchargement

Version actuelle : Version 1.01 (12 avril 2008) ([Lien51](#))

Ce plugin vous est fourni en l'état, sans aucune contrepartie. Si vous souhaitez soutenir son développement, n'hésitez pas à contribuer pour un montant de votre choix via PayPal. Comme toujours, nous vous invitons à contribuer au code pour la correction de bugs et améliorations de fonctionnalités. Quoi qu'il en soit, merci de soutenir nos efforts !

6. Utilisation

6.1. Dépendances

L'explorateur de fichiers jQuery nécessite jQuery ([Lien49](#)) 1.2 ou ultérieurs. Pour les effets, vous aurez besoin du jQuery Easing Plugin ([Lien52](#)) ou d'un autre plugin jQuery ayant le même emploi de votre choix.

6.2. Création d'une arborescence de fichiers

Dans sa forme la plus simple, vous pouvez créer une arborescence de fichiers en utilisant le code suivant :

```
$(document).ready( function()
{
    $('#container_id').fileTree({ root:
    '/some/folder/' }, function(file) {
        alert(file);
    });
});
```

Où **container_id** est l'identifiant d'un élément DIV vide qui existe sur votre page. L'explorateur de fichiers se charge automatiquement lorsque votre page se charge.

6.3. Configuration de l'explorateur de fichiers

Les paramètres sont passés sous forme d'objet à la fonction **fileTree()**. Les options valides comprennent :

Paramètre	Description	Valeur par défaut
root	dossier racine pour l'affichage	/
script	l'emplacement du fichier AJAX à utiliser côté serveur	jqueryFileTree.php
folderEvent	Événement déclenchant Développeur/Réduire	Click

expandSpeed	Vitesse à laquelle les branches s'élargissent (en millisecondes), -1 pour n'utiliser aucune animation	500
collapseSpeed	Vitesse à laquelle se replient les branches (en millisecondes), -1 pour utiliser aucune animation	500
expandEasing	Effet progressif à l'ouverture	none
collapseEasing	Effet progressif à la fermeture	none
multiFolder	Limiter ou non le navigateur à un sous-dossier à la fois	true
loadMessage	Message à afficher pendant le chargement (peut être du HTML)	Chargement...

Pour créer un explorateur de fichiers avec de multiples paramètres, votre code ressemblera à quelque chose comme ceci :

```
$(document).ready( function() {
    $('#container_id').fileTree({
        root: '/some/folder/',
        script: 'jqueryFileTree.asp',
        expandSpeed: 1000,
        collapseSpeed: 1000,
        multiFolder: false
    }, function(file) {
        alert(file);
    });
});
```

6.4. Styliser l'arborescence des fichiers

L'explorateur de fichiers repose à 100% sur du style CSS. Pour que vos utilisateurs apprécient votre page, il faut soit utiliser la feuille de styles fournie pour afficher votre arborescence de fichiers ou créer les vôtres. Reportez-vous à [jqueryFileTree.css](#) pour apporter des changements dans les styles.

6.5. Gérer la réponse

Quand un fichier est sélectionné, l'explorateur de fichiers jQuery passe le nom de fichier en tant que chaîne de caractères. La méthode la plus simple pour capter cet événement est d'utiliser une fonction anonyme. Si vous souhaitez passer le nom du fichier sélectionné à une fonction que vous créez et appelez **openFile()**, votre code ressemblera à ceci :

```
function openFile(file) {
    // do something with file
}
```

```
$(document).ready( function() {
    $
    ('#container_id').fileTree({ [options] },
    function(file) {
        openFile(file);
    });
});
```

7. Scripts serveurur

L'explorateur de fichiers jQuery est livré avec quelques scripts de connexion cote serveurur qui sont utilisés pour lire le système de fichiers sur votre serveur de données et retourner les données au script coté client via AJAX. Le script de connexion par défaut est **jqueryFileTree.php**. Vous pouvez utiliser un script de connexion pour un autre langage par la mise en paramètre du script à l'endroit où vous voulez utiliser le script (voir Configuration de l'explorateur de fichiers). Sinon, vous pouvez coder un script de connexion personnalisé pour étendre les fonctionnalités de l'explorateur de fichiers jQuery afin de mieux répondre à vos besoins.

Les scripts pour les langages suivants sont inclus dans le téléchargement :

- PHP par Cory SN La Viska ([Lien53](#))
- ASP (VBS) par Chazzuka ([Lien54](#))
- ASP.NET (C#) par Andrew Sweeny
- ColdFusion par Tjarko Rikkerink ([Lien55](#))
- JSP par Joshua Gould
- Lasso par Marc Sabourdin ([Lien56](#))
- Lasso par Jason Huck ([Lien57](#))
- Python / Django par Martin Skou
- Ruby par Erik Lax ([Lien58](#))

Si vous souhaitez partager un script de connexion que vous avez écrit pour un autre langage, faites-le nous savoir ! Nous nous ferons un plaisir de l'ajouter à la page de téléchargement pour que tout le monde puisse l'utiliser !

Les scripts fournis avec l'explorateur de fichiers jQuery sont uniquement conçus pour lire des informations auprès d'un dossier racine. Bien que ce soit généralement inoffensif, il existe un potentiel pour des personnes malveillantes de voir l'intégralité de votre structure de répertoire par usurpation des paramètres de la racine. Il est fortement recommandé d'ajouter une certaine forme de contrôle à votre script de connexion pour vérifier que le chemin d'accès lu est un chemin d'accès que vous souhaitez permettre aux visiteurs de voir.

8. Script de connexion personnalisé

Vous pouvez créer un script de connexion pour étendre les fonctionnalités de l'explorateur de fichiers. La méthode la plus simple pour le faire est probablement de modifier l'un des scripts fournis dans le téléchargement. Si vous voulez démarrer à partir de zéro, votre script doit accepter une variable POST(dir) et renvoyer une liste non ordonnée dans le format suivant :

```
<ul class="jqueryFileTree" style="display: none;">
    <li class="directory collapsed"><a href="#" rel="/this/folder/">Folder Name</a></li>
    (additional folders here)
    <li class="file ext_txt"><a href="#"
```

```
rel="/this/folder/filename.txt">filename.txt</a><
/li>
  (additional files here)
</ul>
```

Notez que l'extension de fichier correspondante doit être écrite comme une classe de l'élément **li**, avec le préfixe **ext_**. (Le préfixe est utilisé pour prévenir les noms de classes non valides pour les extensions de fichier qui commencent avec des caractères non-alpha.)

9. Historique des versions

1.01 Mise à jour pour travailler avec des caractères

étrangers dans le répertoire / noms de fichiers (12 avril 2008)

1.00 Version initiale (24 mars 2008)

10. Licence & Modalités d'utilisation

L'explorateur de fichiers jQuery est mise à disposition sous un contrat Creative Commons et les droits d'auteur © 2008 par Cory SN LaViska.

Retrouvez l'article de Cory S.N. LaViska traduit par Alban Lelasseux en ligne : [Lien59](#)



Les débats C++

Le C++ est-il adapté pour le multithreading et la programmation parallèle ?

Bonsoir,

Selon vous, C++ fait-il partie des langages les plus adaptés pour le multithreading et la programmation parallèle, sujet au coeur de la recherche et des évolutions aujourd'hui ?

Quels sont les meilleurs outils pour en tirer au mieux parti ?

N'hésitez pas à argumenter, bien au contraire

PS : voici quelques discussions qui se rapportent au développement parallèle en C++

[\(Lien60\)](#) - [\(Lien61\)](#)

Participez à ce débat en ligne : [Lien62](#)

Les derniers tutoriels et articles

Interview de James Reinders (Intel)

Interview de James Reinders autour d'Intel Parallel Studio

1. Contexte

Cet article est la retranscription d'une interview exclusive que James Reinders d'Intel a bien voulu accorder à Developpez.com. Elle a eu lieu lors de l'Intel Software Conference le 21/04/2009 à Salzburg.

L'interview s'est déroulée par oral et en anglais, les lourdeurs de style que contient cette transcription sont issues d'une tension entre désir de ne pas déformer les propos et besoin de reformuler, j'espère avoir trouvé un dosage acceptable.

Vous trouverez plus de détails sur le reste de la conférence dans cet article [\(Lien63\)](#).

2. Interview

2.1. Questions générales

Quel est votre rôle chez Intel ?

Mon travail comporte deux aspects : Je suis le directeur vente et marketing pour les logiciels de développement Intel, et je suis aussi le principal évangéliste de nos produits. J'ai une formation d'ingénieur, j'aime nos produits et ce qu'on peut en faire, et je donne des conférences, des formations. J'ai aussi écrit des livres (un sur Vtune [\(Lien64\)](#) et un sur les thread building blocks [\(Lien65\)](#)). J'aime garder mes mains dans la technologie, ce qui n'est pas très typique pour quelqu'un faisant du marketing.

Quel est l'intérêt d'Intel pour le logiciel ?

Nous sommes un éditeur et vendons nos produits. Beaucoup de revendeurs distribuent nos logiciels, ça nous rapporte de l'argent, et ça compte beaucoup pour nous. Mais le plus important pour nous reste de fournir un logiciel qui permette à notre matériel de fonctionner. Et je

pense que c'est une bonne chose. On peut être très innovants sur le matériel, et sortir de nouveaux concepts très intéressants, comme SSE4.1 et 4.2 et AVX, mais il est important que l'entreprise qui fournit le matériel fournisse aussi le logiciel qui permet d'en tirer profit. Toute l'industrie en profite.

Et ça nous permet aussi d'avoir un retour. Les développeurs ne le savent peut-être pas, mais ils ont un impact important sur la conception de nos processeurs. Les outils que nous vendons nous donnent une grande expérience dans le support à l'utilisation de nos processeurs par les développeurs.

Ainsi, nos experts en logiciel rencontrent des développeurs qui eux sont experts dans leurs domaines, et nous avons une influence sur la façon dont Intel conçoit les futures générations de processeurs. Ce retour nous évite de développer des processeurs que personne ne pourrait utiliser. Le multi-cœurs, par exemple, est un domaine important, et je suis fier d'aider à en tirer parti. C'est un problème ancien, et on fait des progrès dans la manière de l'aborder.

Parallel studio n'arrive-t-il pas un peu tard, alors que les machines multi-cœurs sont déjà omniprésentes ?

En fait je pense qu'on est dans les temps, voire légèrement en avance. Quand un processeur est disponible, par exemple avec du dual-core, c'est très difficile de convaincre quelqu'un de développer pour ce processeur, quand personne d'autre ne le fait. Désormais, presque tous les processeurs ont au moins 2 cœurs, mais les gains se font sentir de manière bien plus sensible avec 4 ou 8 cœurs, ce qui justifie plus l'effort requis pour que le logiciel en tire partie.

Nous ne sommes pas en retard, car la plupart des développeurs n'ont pas encore développé en parallèle. La majorité du parc installé était encore il y a peu avec un

seul cœur, nous sommes en train de basculer.

Pensez-vous que le développement parallèle va rester confiné à quelques développeurs, ou va-t-il devenir commun ?

Le développement parallèle auquel nous sommes habitué, avec les threads windows ou les pthreads ne sera jamais grand public. Quand on prend un niveau d'abstraction plus élevé, comme ce que permettent les TBB, il devient un peu plus accessible, mais je ne crois toujours pas que tout le monde peut l'utiliser. Mais je pense qu'au fur et à mesure que l'on ajoute des couches plus haut niveau, comme Ct, ou le framework basé sur des acteurs de Microsoft qui peut être intéressant dans certains cas, on progresse.

Et peut-être qu'on se réveillera dans 10 ans, et que tout le monde fera du développement relié au parallèle. Ils ne le feront peut-être pas explicitement, peut-être utiliseront-ils des bibliothèques comme la bibliothèque SQL parallèle de Microsoft, mais leur code sera parallèle.

2.2. Intel Parallel Studio

Parallel studio contient le compilateur C++ d'Intel. Quel est son niveau de compatibilité avec celui de Microsoft. Est-ce qu'il risque d'être un frein à l'adoption du produit ?

Il est *extrêmement* compatible. Aucun compilateur n'est parfait, celui de Microsoft pas plus qu'un autre, il n'implémente pas toujours le standard à la perfection. Nous détectons donc la version du compilateur Microsoft installée, et nous nous comportons de manière aussi proche que possible de celle-ci. Par exemple, et c'est une nouveauté de la version du compilateur dans Parallel Studio, nous générons des messages d'erreur avec les mêmes codes, de manière à être compatible avec des outils comme des filtres qui utilisent ces numéros.

Nous travaillons très dur pour être compatible au niveau du source comme du code compilé. Vous pouvez mélanger librement des modules compilés par ces deux compilateurs : Le name mangling est identique, les tables de code sont les mêmes.

En terme de performance, il peut être intéressant d'utiliser notre compilateur dans Parallel Studio. Il possède par exemple des fonctionnalités comme le support d'Open MP 3.0 (Open MP 2 pour Microsoft). Mais si vous désirez utiliser le compilateur de Microsoft, de nombreux modules de Parallel Studio continueront de fonctionner : Parallel Inspector, Parallel Amplifier, Thread Building Blocks, IPP. Certaines options de debug fournies avec Parallel Composer demandent un code compilé par le compilateur Intel, mais globalement, nous n'imposons pas de compilateur. Finalement, nous avons voulu que le basculement d'un compilateur à l'autre soit immédiat. Il suffit de choisir une option dans un menu.

Votre compilateur est-il entièrement développé par Intel, ou est-il basé sur un front-end spécifique, comme celui d'EDG ?

Nous utilisons le front-end d'EDG depuis plus de 10 ans.

Ils sont vraiment très bons. Nous travaillons aussi en partenariat avec eux pour ajouter à leur produit certaines extensions des compilateurs Microsoft ou GNU, ce qui bénéficie à tout le monde. Le reste du compilateur (back-end, optimisateur...) est développé par Intel.

Votre compilateur est-il à même d'utiliser de lui-même les extensions SSE ou faut-il écrire du code spécifique ?

Notre compilateur est très agressif dans son utilisation de SSE. Lorsque ces instructions ont été ajoutées au processeur, nous avons eu des discussions sur la manière de les concevoir pour qu'elles puissent être utilisées automatiquement par le compilateur, et nous les avons implémentés dans le compilateur en avance de phase afin de vérifier que c'était bien le cas.

Notre compilateur peut déjà profiter d'AVX, le successeur de SSE qui utilise 256 bits au lieu de 128, ainsi que de l'architecture Larabee et ses 512 bits. Bien entendu, on peut toujours aller plus loin dans l'optimisation, mais notre compilateur obtient des gains de temps non négligeables et fait déjà un travail tout à fait correct. Je ne pense pas qu'il existe un autre compilateur qui tire aussi bien parti de SSE que le notre, mais ils essayent...

Les Thread Building Blocks existent depuis un certain temps maintenant, pouvez-vous nous parler des nouveautés de la version fournie avec Parallel Studio, comme par exemple l'utilisation des λ -fonctions ?

Lorsque les TBB sont sorties, nous haïssions le fait que le C++ ne possède pas de λ -fonctions. Ça en rend l'utilisation si peu élégante. Notre équipe voulait presque les ajouter à notre compilateur, et tant pis pour le respect du standard C++. Mais maintenant que le standard C++ les définit, on en profite.

Il y a d'autres évolutions dans les TBB. Déjà, elles sont portées sur un nombre toujours plus important de processeurs. Ensuite, les premières versions supposaient que l'ensemble des tâches allait avancer en permanence, et créaient donc autant de threads qu'il y avait de cœurs disponibles. Ça marche bien pour tout ce qui est calcul, mais moins pour les entrées/sorties, où l'on est souvent en attente. Il est désormais possible de définir une tâche comme étant une tâche d'E/S, et cette dernière sera exécutée dans un thread supplémentaire, afin d'utiliser tous les cœurs de manière optimale.

Quels sont les liens entre TBB et le Concurrency Runtime de Microsoft

Les TBB sont disponibles pour de nombreuses architectures, nous adapterons les TBB pour que sous Windows, elles puissent tourner par dessus le Concurrency Runtime dès qu'il sera disponible. Nous avons déjà demandé à Microsoft d'y ajouter des éléments dont nous avons besoin pour y parvenir.

Les développeurs Linux peuvent-ils espérer développer avec Parallel Studio un jour ?

Le compilateur et les bibliothèques marchent déjà sous Linux. L'année prochaine, nous comptons sortir une

nouvelle version de nos produits (V-Tune, Thread profiler et Thread checker) bénéficiant des avancées que nous avons effectuées dans le cadre de Parallel Studio. Je ne sais pas encore s'ils sortiront sous leur noms actuels, ou s'ils seront renommés Inspector et Amplifier. Beaucoup des technologies les plus avancées prennent naissance sous Linux, nous n'oublions pas cet environnement.

Avez-vous des outils pour d'autres langages que C++ et Fortran ? Certains langages sont ils meilleurs que d'autre pour le parallélisme ?

Inspector et Amplifier travaillent avec n'importe quel langage compilé. Un programme qui mixerait C++ avec par exemple du Java ou du C# pourrait aussi être analysé, mais nous nous concentrons sur le code natif. Je pense que nous allons ajouter quelques fonctionnalités pour aller un peu plus vers C# et peut-être Java, mais ce n'est pas encore défini.

Avec Ct, on se pose la question de faire du parallélisme de données dans d'autres langages, c'est certainement un très bon moyen de démocratiser le parallélisme. Python pourrait être un bon exemple. Nous sommes ouverts aux suggestions sur le sujet.

Pensez-vous que Python soit en train de remplacer Fortran pour l'informatique scientifique ?

D'une certaine manière oui. Les programmes existants en Fortran restent en Fortran, mais pour les nouveaux programmes, Python et Matlab sont très populaires. En particulier, dans les sciences de la vie, où l'usage de l'informatique est plus récent, et où Fortran n'a pas la base installée qu'il peut avoir par exemple en électricité, Python est très populaire.

Le plus important, pour les scientifiques, est de pouvoir écrire dans un langage qu'il maîtrisent, de pouvoir innover, faire évoluer le programme. À partir du moment où ils confient le développement à un développeur, pour des raisons de performance ou de taille du code, et que ce développeur commence à utiliser un langage moins simple, ils perdent cette flexibilité. C'est ce qui fait que Python marche pour ces gens. Et Ct, avec sa notation proche des mathématique pourrait apporter le parallélisme par les données à ces communautés, d'où notre intérêt pour ce langage.

Des outils comme Inspector travaillent en instrumentant l'exécutable et en effectuant des mesures en cours d'exécution. Y a-t-il des outils qui vérifient un programme parallèle de manière statique ?

Notre compilateur peut réaliser quelques tests, mais avec les vérifications statiques, on a le dilemme suivant : Soit on veut détecter tous les problèmes potentiels, mais on aura de nombreux faux positifs, et les gens détestent ça, soit on ne signale que les cas où l'on est certain qu'il y a une erreur, mais on passe à côté de beaucoup de choses. On peut avoir des résultats corrects pour la détection statique des deadlocks, mais pour les data races, qui sont le problème le plus courant, il y a bien trop de faux positifs.

Donc, en pratique, une vérification dynamique est plus

efficace. D'autant plus que nous n'attendons pas qu'une erreur de parallélisme produise vraiment des effets néfastes pour la détecter. Dès que deux threads accèdent au même emplacement mémoire sans utiliser de locks, nous le signalons. La vérification est en pratique limitée aux endroits du programme qui sont exécutés, ce qui demande à l'utilisateur d'exercer l'ensemble de son code, mais ce n'est pas un véritable problème.

Il y a 3 ou 4 ans, nous avons acquis une société qui possédait un autre produit nommé Assurer, qui travaillait en instrumentant le code à la compilation. Quand nous l'avons repris pour sortir Thread Checker, on a décidé d'instrumenter directement le binaire, après la compilation, ce qui permet aussi d'instrumenter des DLL et exécutables pour lesquels on ne possède pas le code source. Quand la première bêta est sortie, nous l'avons diffusée à 15 utilisateurs. J'espérais que parmi eux, 1 ou 2 trouveraient un problème grâce à cet outil. En fait, tous les 15 ont trouvé un data race dans un programme qui était déjà mis sur le marché. Et ces races conditions étaient réelles, elles pouvaient arriver, et elles étaient peut-être déjà arrivées chez certains de leurs clients, provoquant un plantage. Nous avons discuté avec eux, et dans chaque cas, il y avait un conflit entre leur code et du code tiers. Ça m'a vraiment ouvert les yeux sur le fait que les data races sont dans les interactions entre différentes parties du logiciel. C'est un autre problème des vérifications statiques, elles demandent à avoir accès au code source, et passent généralement à côté de ce genre de problèmes.

Avez vous entendu parler du projet de recherche Chess de Microsoft, qui a pour objectif de tester du code multithread en contrôlant comment les différents threads d'exécution s'entrelacent de manière à reproduire les race conditions de manière déterministe ?

En effet, j'ai assisté à une présentation de ce projet au PDC, mais je ne suis a priori pas très convaincu par cette approche. Il essayent de forcer une erreur à apparaître, ce qui est une approche novatrice, mais avec laquelle on n'est jamais certains d'avoir écrit suffisamment de tests. Ce que nous tentons de faire avec Parallel Inspector, c'est détecter la possibilité d'une erreur, même si elle ne s'est pas manifestée au cours de l'exécution.

Un certain nombre de personnes chez Microsoft semblent très excitées par ce projet, donc je suis peut-être passé à côté de quelque-chose. En tout cas, il est très important de rester à l'écoute de toutes les techniques qui peuvent rendre le débogage de programmes parallèles plus simple.

Que pensez-vous des évolutions de C++0x concernant le multithread ?

Avoir un modèle mémoire bien défini rend un langage meilleur, plus portable, mais d'un autre côté, ça complexifie la tâche pour un compilateur de tirer tous les avantages possibles du matériel. Habituellement, chez Intel, les gens proches du matériel aiment bien qu'un langage soit un peu plus ambigu, mais au final, c'est un avantage pour le logiciel que tout soit clairement défini. Je suis certain qu'il y a certains points avec lesquels on peut être plus ou moins d'accord, mais ce qui compte le plus

n'est pas tant la manière dont les choses sont définies, que le fait qu'il y ait une chose clairement définie.

Le C++0x ne contient pas de fonctionnalités pour adresser les deux problèmes du parallélisme : le déterminisme ou le scaling. Ce qu'il fait, et qui est important, c'est fournir un langage plus solide à partir duquel on peut construire des outils qui adressent ces problèmes. Et je suis content qu'ils n'aient pas été distraits en tentant d'ajouter des fonctionnalités haut niveau pour le parallélisme. Je pense que ces fonctionnalités sont plus du ressort d'une bibliothèque comme TBB, PPL or Ct. Je pense qu'il reste beaucoup d'expérimentation à faire avant d'en faire des parties permanentes d'un langage.

Le comité de normalisation du C++ a généralement fait un bon travail en ne standardisant quelque-chose que quand on est certain que l'on en a besoin, et que les compilateurs peuvent l'implémenter. Le comité Fortran, de son côté, a fait des standards qui ne sont pas implémentés en 6/7 ans. Personne n'a un compilateur compatible Fortran 2003, même si on s'en approche.

Que pensez-vous de la bibliothèque pour le parallélisme que Microsoft va inclure dans VC++10, PPL ?

Ma première réaction est que j'aurais souhaité qu'ils utilisent simplement TBB, car je ne pense pas qu'elle résolve un problème différent. Je pense que ces bibliothèques vont être assez similaires, et je pense que l'on pourra faire de TBB un sur-ensemble de ce qu'ils feront. C'est ce que nous comptons faire, ainsi, si on utilise leur bibliothèque, et qu'il manque quelque-chose, il sera possible de basculer vers TBB. Même sur le plan syntaxique, on va tenter d'être très proche d'eux, quitte à fournir une double interface pour certaines fonctions.

Je pense que les gens verront que TBB marche mieux que PPL, mais je n'ai pas de preuves à ce sujet, PPL est dans une phase préliminaire, n'a pas encore été livré, sa définition évolue encore, il faut voir où ça ira. TBB est sur le marché depuis un long moment, beaucoup de gens l'utilisent, et elle a bénéficié de leurs retours. Par exemple, ce dont je vous parlais sur la gestion des E/S dans la nouvelle version de TBB, n'a pas encore d'équivalent pour PPL. Enfin, TBB est portable, ce qui n'est pas prévu pour PPL.

Quelles différences y a-t-il entre la version libre de TBB et celle fournie dans Parallel Studio ?

Techniquement : aucune, c'est le même code source. La version livrée avec Parallel Studio est une version commerciale, avec un support et le code source n'est pas

fourni. La version libre ne bénéficie pas de support de la part d'Intel, vous devez aller dans des forums pour trouver de l'aide. Mais pour être honnête, les gens n'ont pas l'air d'avoir besoin de beaucoup de support pour utiliser TBB.

La version libre est sous licence « GPL with Classpath exception », ce qui signifie qu'on peut se lier avec depuis un programme non GPL. C'est un peu comme la licence LGPL, mais la définition de se lier pour LGPL vient du C, et pour une bibliothèque comme TBB, qui utilise des templates, ce n'est pas suffisant. Mais pour certains clients, c'est important d'avoir une version commerciale, qui ne soit pas GPL, sans cela, ils n'utiliseraient pas TBB. D'autres projets, comme MySQL, utilisent ce schéma où le même code source est livré sous un double système de licence.

Quoi qu'il en soit, la version commerciale et la version libre fonctionnent de manière identique.

Pourriez vous nous parler un peu plus de Ct ?

Si vous allez sur le site ([Lien66](#)) vous pourrez avoir plus d'informations sur ce sujet. Il y a un certain nombre de whitepapers sur le sujet, et certains ont des exemples de code source.

Le principe de base est assez simple. On utilise la surcharge d'opérateur pour déclarer un vecteur et être capable de dire $a=b*c$. On peut faire nombre de calculs avec ces vecteurs, mais on n'a pas le droit d'avoir un pointeur qui pointe sur le contenu de ces vecteurs. Cette restriction nous permet de savoir que seul Ct manipule le vecteur, et nous autorise donc à paralléliser le code de manière déterministe. Ces vecteurs peuvent être pleins ou creux, ce qui est souvent nécessaire pour les calculs mathématiques, mais cet aspect est totalement géré et encapsulé par notre bibliothèque.

Est-ce une technique semblable aux expressions template, comme par exemple dans Blitz++ ?

Probablement. Par exemple, les calculs ne sont pas exécutés immédiatement, mais de manière paresseuse, de manière à agréger certains calculs. Ainsi, quand on fait $a=b+c+d$, on attend d'avoir l'ensemble de l'expression pour faire le calcul de manière efficace, sans exécuter boucle sur boucle sur boucle, et cette agrégation peut avoir lieu sur plusieurs instructions. Ça réduit les déplacements mémoire, et c'est très bon pour le parallélisme, car on peut avoir une vision d'ensemble, et ainsi mieux paralléliser. Et l'avantage, c'est que tout ça a lieu en restant du pur C++.

Retrouvez l'article de Loïc Joly en ligne : [Lien67](#)

Test d'Intel Parallel Studio (version bêta)

J'ai testé le dernier-né des outils Intel, Parallel Studio ([Lien68](#)).

1. Introduction

Intel Parallel Studio est un ensemble d'outils dédiés à l'optimisation des programmes multithreadés. Il s'agit de

plusieurs plugins de l'environnement Visual Studio. Il est donc nécessaire de posséder ce dernier (attention, la version Express ne supporte pas les plugins).

On trouvera donc les plugins suivants, pouvant aussi être achetés séparément :

- Parallel Inspector, analysant la gestion de la mémoire
- Parallel Amplifier, analysant le comportement des threads
- Parallel Composer, contenant l'extension parallèle pour le débogueur

2. Les outils connus

2.1. Le compilateur C++ 11.1

Pour le moment disponible en version bêta, la version 11.1 du compilateur d'Intel propose de nouvelles fonctionnalités :

- OpenMP 3.0
- Intégration de TBB et IPP

Je ne vais pas faire une analyse complète de ce compilateur, cela peut se résumer à d'excellentes performances, des extensions utiles et un support de la norme C++0x telle qu'elle est définie actuellement (par exemple les fonctions lambda).

2.2. Threading Building Blocks

Il s'agit d'un ensemble d'outils permettant de découper un travail en bloc, chaque bloc étant envoyé sur un thread. La répartition est effectuée par un ordonnanceur.

2.3. Integrated Performance Primitives

Les outils de traitement du signal se trouvent dans IPP. Il s'agit de fonctions non standard (contrairement à celles proposées dans la MKL, la bibliothèque scientifique optimisée d'Intel, malheureusement absente) pour effectuer des traitements 1D, 2D, sur des vidéos, ...

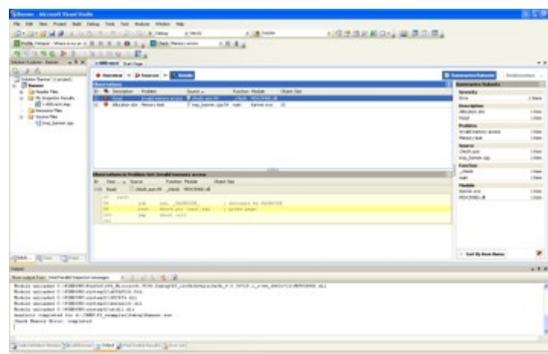
3. Les nouveaux outils

Chacun des 3 plugins du Parallel Studio peut s'utiliser séparément. Il y a peu d'exemples, malheureusement, aucun n'est commun à plusieurs plugins, et ils sont plus ou moins simples. Au moins, on peut voir ce que les plugins font, mais les limites ne peuvent pas être abordées.

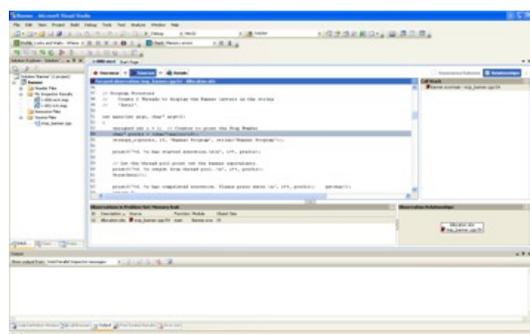
3.1. Parallel Inspector

L'inspecteur est chargé de détecter les problèmes de mémoire et de threads lorsqu'un programme fonctionne. Selon le niveau d'inspection, le temps d'exécution du programme peut être très long (mais on n'a rien sans rien). A chaque fois qu'un problème est détecté, sa gravité sera indiquée, l'emplacement de l'erreur et le code associé (qui pourra ensuite être modifié dans l'éditeur).

La première analyse possible est celle de la mémoire. Il est donc possible de vérifier, entre autres, les fuites mémoire. Le rapport reporte l'endroit d'allocation. Tout ceci est visible dans la capture d'écran suivante.

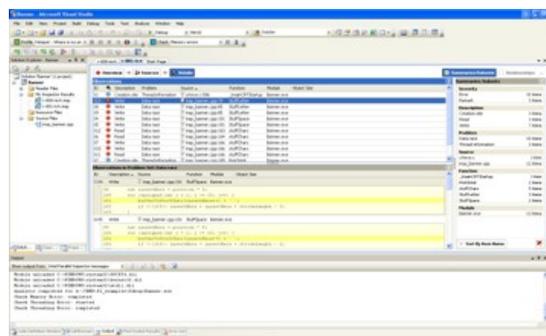


Rapport d'inspection mémoire

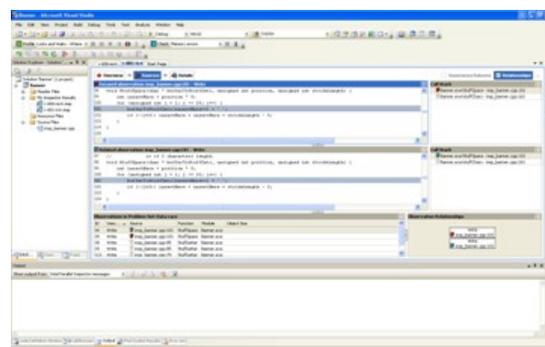


Le code source ayant généré une fuite mémoire

Parallel Inspector n'est pas Parallel pour rien. L'inspection des threads permet de vérifier les accès mémoire concurrents. Il est possible de supprimer certaines analyses afin d'accélérer l'inspection (mais seulement après qu'une première inspection ait été effectuée). Par exemple, si on connaît le comportement d'une variable, il n'est pas nécessaire de la surveiller.



Rapport d'inspection sur le parallélisme



Affichage d'un problème d'accès en écriture à une même donnée

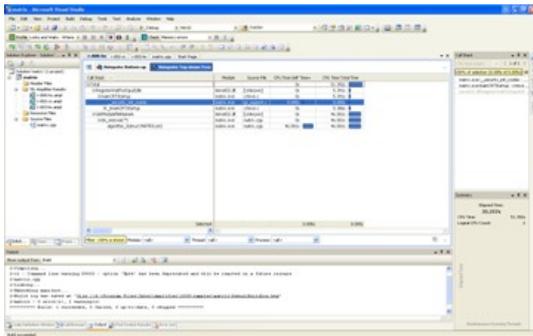
L'analyse mémoire est un problème récurrent. Il est parfois nécessaire d'ajouter une bibliothèque lors de la compilation ou de modifier son code. Ici, tout est fait en direct, ce qui est bien pratique. De même, l'inspection des

threads est un outil qui peut être très utile lors d'un débogage complexe où le multithreading intervient.

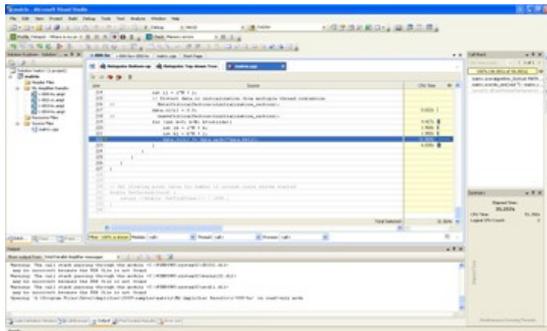
3.2. Parallel Amplifier

Amplifier effectue un profil du code à la volée. Le profil d'un code peut déjà être effectué par certaines versions de Visual Studio ([Lien69](#)), mais avec plusieurs coûts différents. Ici, on se limitera à la durée de fonctionnement, à la qualité du parallélisme et aux délais d'attentes.

Le premier type de profil est le **hotspot**. Il s'agit de mesurer où l'application passe le plus clair de son temps. Dans le cas présenté, c'est la fonction **algorithm3** qui est la plus coûteuse. En double-cliquant dessus, on ouvre une fenêtre donnant ligne par ligne le pourcentage de temps passé.

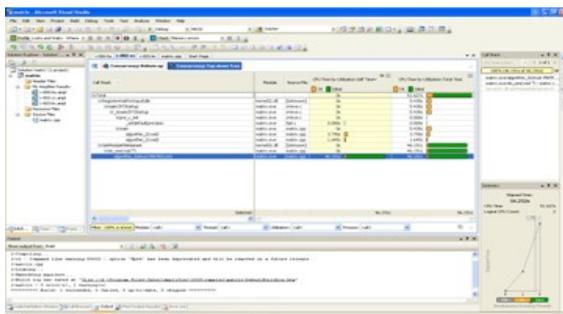


Détection des fonctions les plus coûteuses

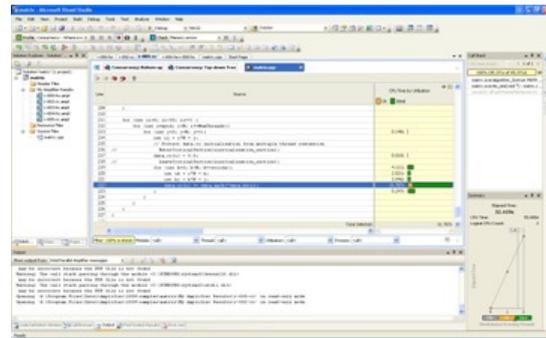


Détail des coûts par instruction

L'analyse du parallélisme, **concurrency**, indique, à partir du nombre de processeurs de la machine, la scalabilité de l'algorithme. Dans le cas présenté, on voit en bas à droite le gain total, pour deux processeurs qui est à 1.78, soit donc 88.8% d'utilisation. On peut demander le détail des sources qui indique que le manque de parallélisme vient surtout des routines d'affichage. En revanche, la fonction la plus coûteuse, *algorithm3*, en temps a un bon parallélisme.

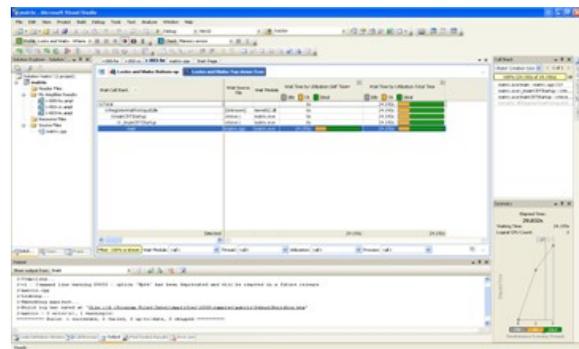


Détection du parallélisme sous-optimal

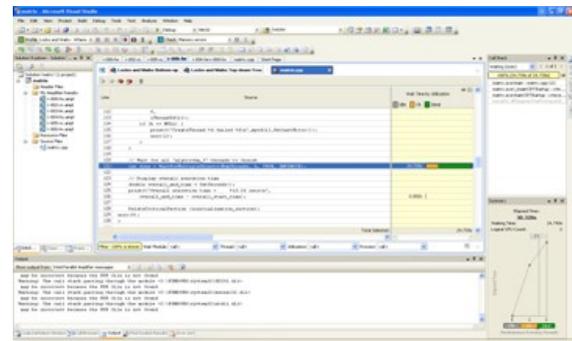


Détection du parallélisme sous-optimal

Enfin, un des gros problèmes du parallélisme concerne les attentes et les verrous. Ici encore, Amplifier propose une solution. Dans l'exemple ici, l'attente n'est présente que dans la fonction principale, lorsqu'elle attend que les sous-threads se terminent.

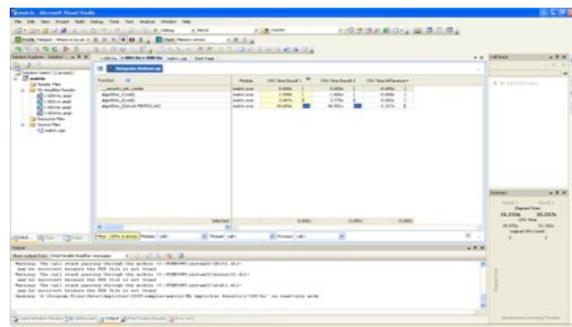


Détection des attentes et des verrous coûteux



Détail des verrous par instruction

Il est aussi possible de comparer les résultats entre plusieurs exécutions et changements. Cela permet de visualiser l'évolution des améliorations.



Comparaison entre deux exécutions hotspot

Les données retournées sont très pratiques et utiles (comme la vue sur le code source), mais il manque

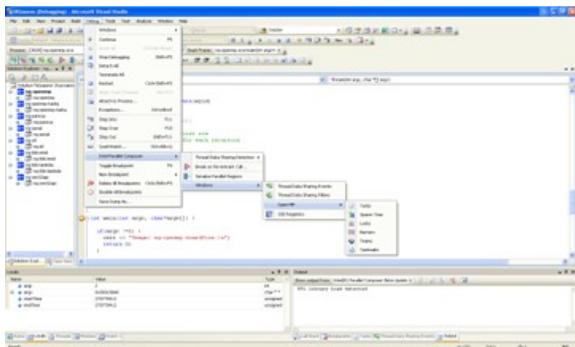
décidément de la documentation. L'aide en ligne ne donne pas suffisamment d'informations, et la signification du cadre en bas à droite est incomplète. Par exemple, à quoi sert la petite flèche entre les indications "OK" et "ideal" ? Le tooltip ne le dit pas...

3.3. Parallel Composer

La documentation de cet outil est sans doute la plus complète. Malheureusement, elle est dispersée dans différents dossiers, sans forcément porter des noms de dossiers très explicites (showme pour l'animation Flash de démonstration).

L'ajout principal de ce plugin est l'extension parallèle pour le débogueur. C'est aussi avec ce plugin qu'est livré le compilateur Intel 11.1. Il y a donc des liens forts entre les deux. Pour activer l'extension, il est nécessaire d'indiquer que l'on désire détecter les événements parallèles (ce qui permet de ne pas toujours avoir ce surcoût lorsqu'on débogue une application classique).

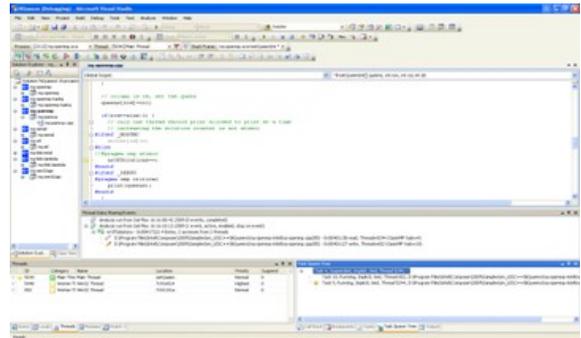
L'intérêt de cette extension n'est pas que dans la détection de données partagées entre threads (Inspector le fait aussi, mais ne retourne pas l'information dès qu'elle est disponible comme c'est le cas pour un débogueur), il s'agit aussi de vérifier des comportements dynamiques comme le code réentrant (est-ce qu'une fonction peut être appelée simultanément par plusieurs threads ?) ou l'arbre des tâches et des threads avec OpenMP.



Possibilités offertes par l'extension en mode debug

Les sous-fenêtres OpenMP ne sont pas que pour OpenMP. A partir du moment où l'option `/qopenmp` est utilisée (elle est nécessaire pour OpenMP, mais aussi pour les extensions parallèles du compilateur comme `__par`), elles peuvent afficher des informations utiles sur l'état des

threads existants. De plus, il est alors possible de basculer en exécution monothread, ce qui permet aussi de vérifier le séquençement des exécutions.



Différentes vues en cours de débogage, comme l'arbre des tâches ou les événements

Il semblerait qu'il soit possible de déboguer plusieurs processus simultanément, à la TotalView ([Lien70](#)), mais aucun exemple ne le montre, aucune aide n'en parle, donc je suppose que ce n'est pas accessible dans la bêta et qu'un exemple non existant pour l'instant sera proposé.

C'est sans doute le plugin le plus important dans Parallel Studio, mais je trouve qu'il reste un goût de documentation et tutoriels inachevés. Il manque des exemples plus frappants de l'utilité de cet outil.

4. Conclusion

Comme je l'ai indiqué dans les derniers paragraphes de chaque mini-test, si Amplifier et Inspector sont intuitifs et simples d'utilisation, ce n'est pas le cas de Composer. Il manque clairement des exemples et des tutoriels (la seule vidéo n'est utile que pour commencer à utiliser Composer, pas pour montrer son efficacité). Cela va sans doute s'arranger dans les mois qui viennent, mais c'est dommage pour un produit qui promet, mais qui déçoit à ce niveau.

Il n'empêche que c'est un produit qui me semble très utile, qui est peu envahissant dans son programme (je pense à Inspector et Amplifier pour la détection de certains problèmes sans bibliothèques additionnelles), et efficace. Les problèmes qu'il tente de résoudre ne sont pas faciles, et il s'en sort, à mon avis, avec brio. Espérons que la documentation et l'aide seront améliorées !

Retrouvez l'article de Matthieu Brucher en ligne : [Lien71](#)

Initiation à l'utilisation et au paramétrage d'un site SharePoint

Ce tutoriel a pour but de vous initier à la conception et gestion d'un site SharePoint à travers son paramétrage et son utilisation. Ce tutoriel s'adresse aux débutants, gestionnaires de sites ou plus particulièrement de publication de contenu.

1. Introduction

Ce tutoriel, destiné aux débutants et gestionnaire de contenu de sites, permet d'aborder des notions de bases concernant la compréhension globale d'un site SharePoint et tout ce qui se rapporte au paramétrage. Il permet d'avoir un aperçu des fonctionnalités et possibilités de SharePoint mais n'a pas la prétention de tout regrouper.

Nous découvrirons les sites SharePoint en 2 phases.

La première phase est composée d'une présentation et définition d'un site SharePoint, ses objectifs et intérêts, ses principales fonctionnalités (afin de découvrir l'étendue des possibilités de MOSS 2007) et son architecture technique (qui permet d'appréhender le fonctionnement général et les interactions possibles). De même, nous aborderons une présentation de l'environnement SharePoint, plus particulièrement une page type et tout ce qui se rapporte à la sécurité et gestion des droits utilisateurs.

Après avoir découvert l'environnement et quelques fonctionnalités, la seconde phase réfère à tout ce qui se rapporte à la gestion de sites, gestion de pages web, gestion de contenu comme les bibliothèques ou les listes.

Remarque : Les explications fournies sont réalisées sur Microsoft Office SharePoint Server 2007 (MOSS 2007). Cependant, de nombreuses fonctionnalités se retrouvent dans Windows SharePoint Services (WSS) qui est un add-on gratuit de Windows Server 2003.

2. SharePoint, pour qui? Pourquoi?

2.1. Définition

Un site SharePoint est par définition un point de partage, utilisant tous les avantages du web, car c'est un site web qui permet de partager des informations en générales, des documents, des plannings, etc. Il permet d'un point de vue plus global d'améliorer la communication au sein d'un système d'information et de faciliter tous les échanges. Un site SharePoint, d'un point de vue utilisateur ou gestionnaire de contenu, est facile d'utilisation, fournit une interface agréable sans aucun développement, un contenu dynamique et responsabilise les utilisateurs. En effet, en fonction des droits, l'utilisateur peut lui même apporter une information et participer à la vie du site.

Remarque : Cette définition n'est pas tout droit sortie d'un dictionnaire mais de mes impressions, par conséquent elle

n'est pas à prendre au pied de la lettre et reste à enrichir par votre propre expérience.

2.2. Objectifs et intérêts principaux d'un site SharePoint

Les outils SharePoint permettent à n'importe quelle personne (avec un site existant ou bien le matériel et logiciels nécessaires) de construire un site web sans aucun développement. De nombreuses fonctionnalités répondant à la plupart des besoins en entreprise sont existantes, tout ce qui se rapporte à la gestion du site, du contenu, de l'interface, etc. est paramétrable. L'intérêt est donc d'utiliser au maximum ces fonctionnalités qui couvrent énormément de besoin. Parfois celles-ci ne sont pas suffisantes. Dans ce cas un développement spécifique est nécessaire mais dépasse largement le niveau de ce tutoriel.

Avec un site SharePoint, il est possible d'accéder à une information à travers une simple URL et un navigateur web, d'avoir des informations vivantes, qui évoluent, un temps de réaction moins important au niveau du transfert de l'information, etc. En plus d'offrir un accès facilité à l'information, SharePoint est un véritable outil de travail, qui s'associe à la suite Office et tous ces avantages : feuilles de calculs en ligne, reporting, planning et tâches, etc.

En résumé, ce type de site simplifie le travail, la prise de décision, permet la collaboration et s'adapte à tous les métiers et presque tous les besoins.

3. Principales fonctionnalités

Ceci est une présentation exhaustive des possibilités de SharePoint mais permet de se faciliter la vie quant aux questions préliminaires sur la faisabilité.

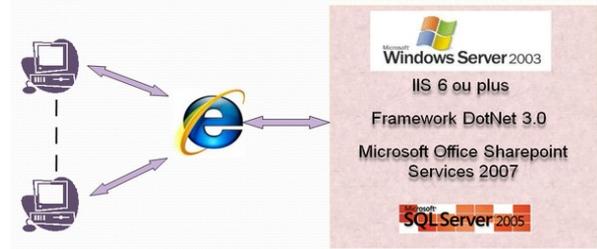
Fonctionnalités	Détail
Création de site(s), création de page(s)	SharePoint permet de créer ses propres sites, ou pages très aisément, en précisant un nom, en choisissant un modèle de site ou de page, etc. Des modèles permettent de pré-remplir les pages ou de proposer des composants qui faciliteront l'obtention du résultat escompté. Un site peut être enregistré en tant que nouveau modèle de sites.

Gestion des utilisateurs, groupe d'utilisateurs	Gestion des profils utilisateurs, groupes, gestion des droits par domaines NT, groupes ou utilisateurs. Ce type de gestion des droits permet de gérer la visibilité pour un/des utilisateurs au niveau : sites, onglets, listes, bibliothèques, répertoire, documents. Par défaut, les droits sur un site enfant sont les mêmes que sur son parent. Ceci est modifiable.
Moteur de recherche	SharePoint permet de mettre en place un moteur de recherche très puissant. Il permet de rechercher dans le contenu des documents stockés dans SharePoint, dans les métadonnées associées aux documents ou des profils utilisateurs. Sa gestion s'effectue dans la centrale d'administration. Les résultats sont aussi filtrés en fonction des droits utilisateurs, tous les sites peuvent ne pas être dans la liste de recherche, etc.
Statistiques d'utilisation du site	Permet d'obtenir des statistiques concernant les visites du site SharePoint, le nombre de visites par jours, les mots les plus recherchés, les sites les plus utilisés, etc.
Gestion documentaire	Avec résultats filtrés fonction des droits de l'utilisateur, préférences des résultats affichés paramétrables, abonnement flux RSS possible : notification par email, gestion des versions, gestion des validations, gestion de l'accès concurrent à un document, etc.
Gestion des tâches et calendrier équipe	Composant de type calendrier Outlook, synchronisation avec Outlook.
Gestion planning	Gantt project
Excel services	SharePoint permet de retrouver toutes les fonctionnalités d'Excel directement dans le site. Les import/export Excel-site et site-Excel sont possibles.
Collecte d'informations	Avis des utilisateurs, sondages, enquêtes, aperçu graphique des résultats.
Reporting	Avec l'intégration d'Excel et reporting services (SSRS) d'SQL Server.

Intégration de template (modèle) documentaire	Grace à l'intégration d'InfoPath.
---	-----------------------------------

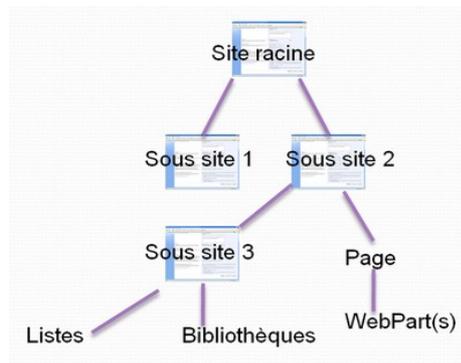
4. Architecture SharePoint

4.1. Architecture technique



Remarque : Afin d'accéder à un site SharePoint, l'utilisateur a besoin d'un navigateur web ainsi que de Microsoft Office pour profiter de toutes les fonctionnalités que propose MOSS 2007. Concernant le serveur, il nous faut un Windows 2003 server pour le système d'exploitation avec IIS 6 ou plus et ASP.net 2.0 activé dans les extensions de IIS. Il nous faut aussi installer le Framework DotNet 3.0, un Microsoft SQL Server 2005 et Microsoft Office SharePoint Server 2007 (MOSS 2007).

4.2. Contenu et structure d'un site

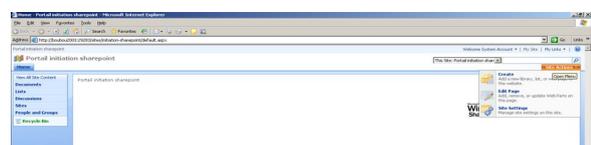


Comme nous pouvons le voir, voici un exemple de **collection de sites** avec site racine, et plusieurs sous sites possibles.

5. L'environnement SharePoint : une page type

Nous allons découvrir l'environnement d'une page type et les premières possibilités offertes grâce aux diverses actions du menu (lorsque l'on dispose déjà d'un site créé). Une page se divise en trois parties principales :

1. Le menu à gauche : qui permet d'accéder à tout le contenu du site.
2. La page au centre : le corps de la page elle-même.
3. Le **menu action du site** permet d'atteindre 3 menus : **créer, modifier la page et paramètres du site**.

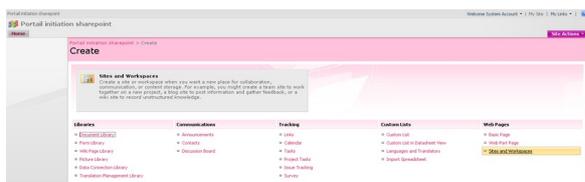


- Le menu **Actions du site** est visible lorsque l'utilisateur est connecté et donc authentifié, et a

suffisamment de droits pour le visualiser. Il permet de créer des pages, bibliothèque ou liste. Il permet de modifier la page à savoir ajouter, retirer ou modifier des composants WebPart (dans ce cas le site précise que nous sommes en mode édition). Enfin, paramètres du site, permet de gérer tout ce qui se rapporte aux paramètres du site (comme mettre le thème du site en rose :p) :

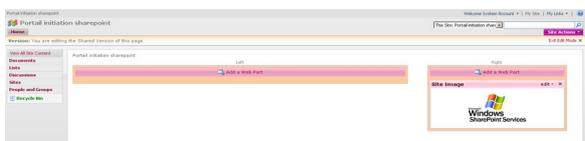


Concernant le menu **créer** :



Nous pouvons voir que la création consiste en 4 parties : bibliothèques, communications, suivi, listes personnalisées et page web.

Concernant le menu **modifier la page** :



Lorsque l'on souhaite modifier la page courante, nous passons en mode édition. Nous pouvons voir les différents espaces d'ajout de webpart.

- Le menu **Afficher tout le contenu du site** permet de visualiser la structure du site. Il permet de travailler sur un élément du site comme une liste, créer une bibliothèque, un forum de discussion, une enquête, une page, un sous-site. Pour les créations modification, soit nous passons par ce menu, soit toujours par le menu **Actions du site**.



Ce menu permet de visualiser la structure du site courant. Dans notre cas, nous pouvons constater que le site ne contient rien car il a été créé spécifiquement pour ce tutoriel, nous le modifierons au fur et à mesure.

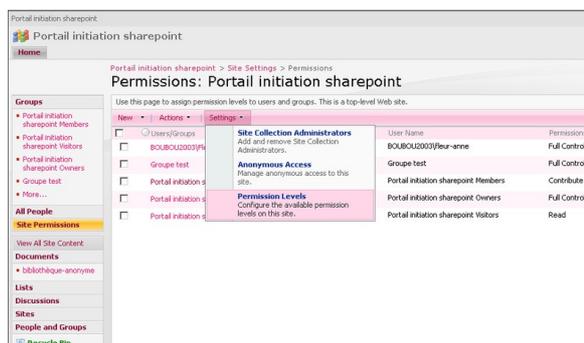
6. Gérer les niveaux d'autorisations

Nous allons débiter cette partie par la notion de niveau d'autorisation car par la suite ces niveaux d'autorisation sont applicables à des groupes d'utilisateurs, des

utilisateurs seuls, des bibliothèques et listes, des dossiers et des documents.

6.1. Définition et description des niveaux d'autorisations

Un niveau d'autorisation correspond à un ensemble d'autorisations qu'il est possible d'attribuer à des utilisateurs et/ou groupes d'utilisateurs ou plus en détail à un site, une liste ou bibliothèque, un dossier et même un document, fonction des stratégies de sécurité souhaitées. Par défaut, il existe 5 types de niveaux d'autorisations. Ceux-ci sont accessibles via les paramètres du site, soit **Actions du site>Paramètres du site>Permissions avancées puis Paramètres> Niveaux d'autorisations**. Nous accédons alors à cet écran listant les niveaux d'autorisations existants (notons qu'il y en a qui sont grisés) :



- **Le contrôle total** : Ce niveau d'autorisation n'est pas modifiable ni supprimable. Celui-ci est accordé par défaut au propriétaire du site.
- **Droit de création** : Ce niveau d'autorisation permet d'effectuer tout ce qui se rapporte à la création de bibliothèque ou liste et de modifier des pages.
- **Droit de collaboration** : Ce niveau d'autorisation permet de contribuer au site à travers de la création, modification et suppression d'élément de bibliothèque ou liste (dossier ou document) ou encore de créer des pages web. Par défaut, cette autorisation est affectée à un groupe appelé **Contributeurs+nom du site**.
- **Droit de lecture** : Ce niveau d'autorisation permet uniquement d'accorder un droit de lecture sur le site. Par défaut, ce niveau d'autorisation est associé au groupe appelé **Lecteurs+nom du site**.
- **Accès limité** : Ce niveau d'autorisation accès limité est un niveau non supprimable ni modifiable. Ce niveau d'autorisation est fait lorsque l'on veut ne donner accès qu'à un document précis, une bibliothèque précise. Attention, si ce niveau est appliqué à un groupe ou un utilisateur, il faut aussi donner les accès à ce même groupe ou utilisateur à la bibliothèque contenant le document en question. Avoir les autorisations à une information sans avoir les autorisations de son contenant ne permet pas d'y accéder.

Si un sous-site hérite des autorisations de son parent, la gestion des autorisations du sous-site n'est pas possible. Soit il faut passer par le site parent et modifier les

autorisations (ce qui est risqué pour tous les autres sites enfants au niveau de la stratégie de sécurité choisie), soit rompre l'héritage des autorisations du site enfant.

6.2. Gestion des niveaux d'autorisations : création, modification, suppression

Pour créer un niveau d'autorisation, il nous suffit de reprendre le chemin précédemment décrit par les captures d'écran à savoir : **Actions du site>Paramètres>Autorisations avancées**. Ensuite, nous faisons **Paramètres>Niveaux d'autorisations**. Et à ce niveau la, nous pouvons faire Ajouter un niveau d'autorisation.



Pour ajouter un niveau de permissions 2 catégories de renseignements sont demandés :

- **Nom et description**
- **Permissions**

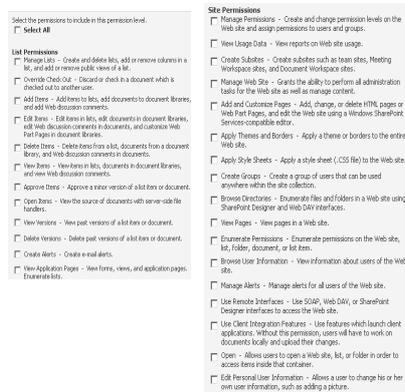
Concernant le **nom et la description** rien de plus simple il suffit de renseigner. Dans notre cas, nous allons créer un niveau d'autorisation groupe test que nous affecterons par la suite.



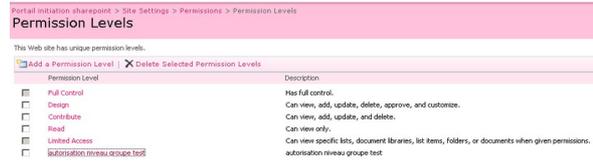
Concernant les permissions, elles se divisent elles-mêmes en 3 catégories :

- **Autorisations de liste**
- **Autorisations de site**
- **Autorisations personnelles**

Il nous suffit de combiner les autorisations souhaitées pour ces 3 catégories. Ci-dessous, un aperçu des autorisations :



Nous validons les autorisations choisies et notre nouveau niveau d'autorisation est créée.



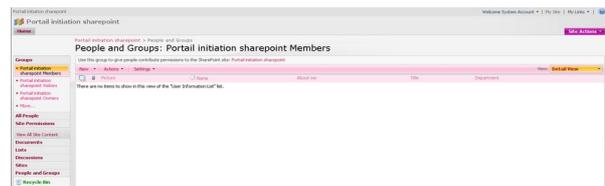
Remarque : Certaines autorisations dépendent d'autres. SharePoint les coche automatiquement lorsqu'une autorisation a une dépendance avec une ou plusieurs autres. Par contre, si l'on décide de revenir en arrière, SharePoint ne décoche pas les dépendances (bien faire attention s'il faut les supprimer pour revenir en arrière). Ou bien il nous faut faire annuler.

Pour modifier un niveau d'autorisation, il suffit de rester dans la fenêtre précédente et de sélectionner le niveau à modifier.

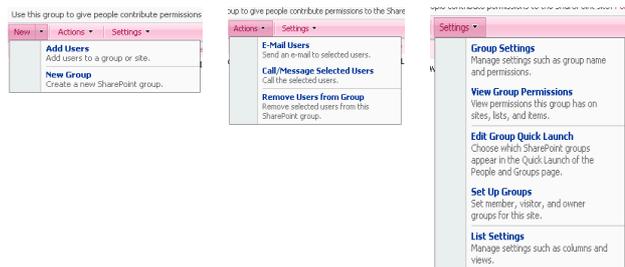
Enfin, pour supprimer un niveau d'autorisation, il nous faut toujours rester sur cette même fenêtre où nous pouvons visualiser les différents niveaux d'autorisation possibles, en sélectionner un et cliquer sur supprimer les niveaux d'autorisation sélectionnés.

7. Gestion des groupes : création, modification, suppression

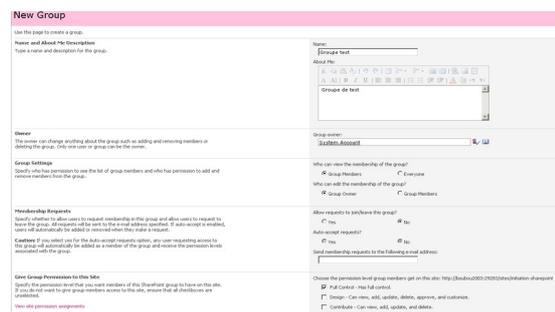
Après avoir sélectionné **Personnes et groupes**, nous obtenons ce type d'affichage.



Nous pouvons constater que notre menu s'est modifié. Nous pouvons voir les groupes existants, qui se créer par défaut : membres, visiteurs et propriétaires. De même, un nouveau menu apparaît. Ce menu va nous permettre d'accéder à la création, modification, propriétés, etc.



Commençons par **créer un nouveau groupe**. Pour cela, il nous faut faire **nouveau>nouveau groupe**.



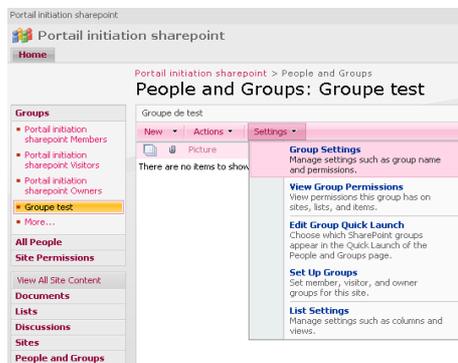
Un groupe se caractérise par un nom et une description (facultatif), un propriétaire, des paramètres, des autorisations concernant les demandes d'accès et enfin des permissions. Il existe plusieurs types d'autorisations :

- Contrôle total
- Conception : permet aux membres de créer des listes, bibliothèques et modifier des pages du site
- Collaboration : permet aux membres de créer, modifier le contenu de listes et bibliothèques du site
- Lecture seule

Nous validons la création de notre nouveau groupe (groupe test). Nous pouvons constater que le groupe a bien été créé car celui-ci apparaît alors dans le menu de navigation :



Pour modifier un groupe, il suffit de se rendre sur **Personnes et groupes>Groupe test** dans notre cas, puis faire **Paramètres>Paramètres du groupe**. Nous retombons alors sur la même fenêtre qu'à la création d'un groupe où il suffit de valider ou d'annuler toutes modifications.

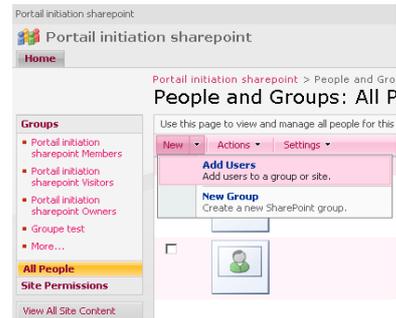


Pour affecter un niveau d'autorisation spécifique (par exemple celui que nous avons créé précédemment) à notre groupe test, il suffit d'éditer les permissions du groupe choisi (**Permissions de site > Groupe test**). Puis, il nous est possible de choisir le niveau d'autorisation que nous avons créé précédemment :

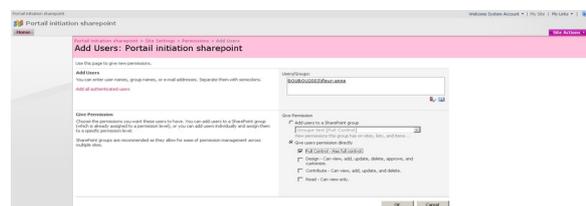


8. Gestion des utilisateurs : création, modification, suppression

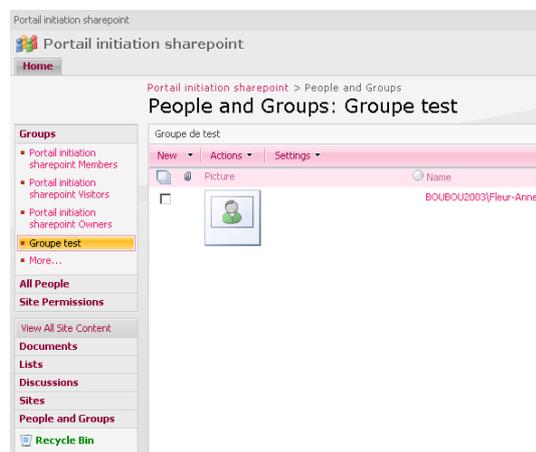
Pour ajouter un utilisateur, nous devons faire **New>Ajouter un utilisateur** que ce soit au niveau du site ou d'un groupe sélectionné :



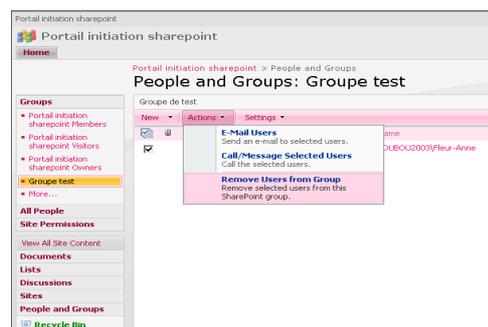
Ci-dessous, voici les informations à renseigner lors de l'enregistrement d'un utilisateur dans un groupe :



Nous pouvons constater qu'il faut renseigner la personne à ajouter bien entendu mais qu'il est possible soit de l'inclure dans le groupe avec les droits du groupe, soit de lui donner des droits spécifiques. Comme nous pouvons le constater, l'utilisateur créé est bien dans le groupe test que nous avons créé spécifiquement.



Pour supprimer un utilisateur, il suffit de le sélectionner (cocher la case) et de faire **Actions>Supprimer un utilisateur d'un groupe** comme suit :



9. La connexion anonyme

La connexion anonyme est un paramètre à utiliser avec précaution et nous allons expliquer tout simplement pourquoi ci-dessous.

Tout d'abord, activer la connexion anonyme, cela veut dire que les utilisateurs anonymes et ceux ne disposant pas normalement de droits d'accès sur le site web peuvent tout simplement parcourir tout le site. Parcourir tout le site signifie pouvoir lire les listes, les bibliothèques, tous documents sur le site, etc. (si ceux-ci héritent des autorisations du site). Lorsque l'accès anonyme est activé (par l'administrateur du serveur), le propriétaire du site peut alors :

- Activer l'accès anonyme à tout un site
- Activer l'accès anonyme qu'a des listes et bibliothèques du site
- Refuser tout accès

Voyons les droits accordés aux utilisateurs anonymes ainsi que ceux non autorisés.

- **Possibilités et droits en accès anonyme**

Un utilisateur en connexion anonyme peut parcourir tout le site, lire des bibliothèques, des listes, répertoire lorsque ces mêmes éléments héritent des droits du site, en particulier lorsqu'il a été précisé que l'affichage des éléments est accordé aux utilisateurs anonymes.

- **Impossibilités en accès anonyme**

Un utilisateur en connexion anonyme ne peut pas par défaut modifier le contenu du site, d'une liste, bibliothèque ou autre disposant des mêmes autorisations que le site.

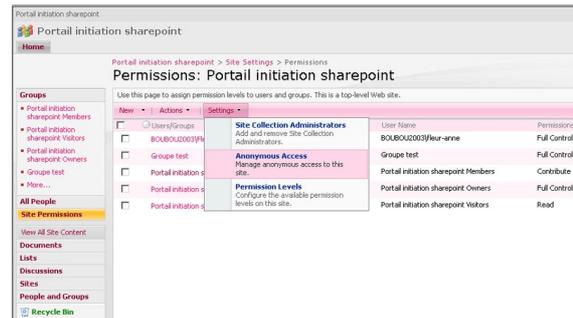
Remarque : Il est bien évident qu'activer l'accès anonyme et donnant un contrôle total ne permettra pas d'historiser, ni de retracer les propriétaires de chaque action effectuée sur le site. De plus, un utilisateur anonyme disposant de tous les droits peut lui même accorder un accès anonyme à un site, modifier les droits, etc. sans que la personne soit

identifiée. Pour les sites plus sécurisés, il est déconseillé d'activer ce type de connexion.

9.1. Activer l'accès anonyme sur un site

Comme précisé précédemment, si l'administrateur n'a pas préalablement activé cette fonctionnalité au niveau de l'application web, l'option d'accès anonyme n'apparaît pas dans les paramètres.

Pour activer la connexion anonyme au niveau d'un site, il suffit de nous rendre à **Actions du site > Paramètres du site**. Puis, au niveau des **utilisateurs et permissions**, nous cliquons sur **permissions avancées**. Enfin, au niveau des paramètres, nous pouvons voir l'**option accès anonyme** (si l'administrateur l'a effectivement activé), comme suit :



Ensuite, nous arrivons sur une fenêtre importante dans laquelle nous devons définir les droits que nous souhaitons accorder :



Il est possible de donner les droits en anonyme à tout un site, uniquement aux listes et bibliothèques ou enfin aucun droit.

Retrouvez la suite de l'article de Fleur-Anne Blain en ligne : [Lien72](#)

Des disques durs lus/écrits 100 000 fois plus vite grâce à des lasers !

Bonjour,

Selon les résultats de travaux de l'équipe de M. Jean-Yves Bigot (Institut de physique et chimie des matériaux de Strasbourg), des impulsions lasers ultracourtes permettent de multiplier par 100.000 la lecture et écriture sur les disques durs !

À l'heure actuelle, pour lire et écrire des données sur un disque dur, le procédé utilisé est la polarisation des électrons sur les têtes de lecture, le nouveau procédé quand à lui utilise les photons (des particules de lumière) émis par les lasers.

M. Bigot a par ailleurs expliqué :
« Avec l'écriture à tête magnétique, on parle de spintronique, ou d'électronique de spin. Notre méthode, c'est la photonique de spin, car ce sont les photons qui modifient l'état d'aimantation des électrons »

Ainsi que :
« On ne savait pas bien quelle était la limite de temps pour

l'inscription. Nous venons de montrer qu'elle a lieu pendant la durée de l'impulsion du laser, soit un millionième de milliardième de seconde »

Pour information, millionième de milliardième de seconde est plus couramment appelé femtoseconde.

Cependant cette technologie n'est pas encore destinée au grand public car les lasers utilisant ces procédés sont d'une dimension assez conséquente, à savoir, 30cm sur 10cm...

Toujours d'après M. Bigot :
« Il faudrait descendre à l'échelle du centimètre pour pouvoir être sur une puce électronique. Dans les dix ans qui viennent, on devrait avoir des lasers femtoseconde, ou au moins à la picoseconde à cette échelle »

Inutile de dire que les grands fabricant sont déjà sur le qui-vive et intéressé par cette nouvelle technologie.

Très prometteur donc, ne trouvez-vous pas?

Site officiel : IPCMS Strasbourg ([Lien73](#))

Commentez cette news en ligne : [Lien74](#)



Guide des meilleures astuces pour Mac

- Comment retrouver un fichier perdu ?
- Comment forcer une application à quitter ?
- Comment déplacer son dossier Musique sur un disque externe ?
- Comment convertir une vidéo dans un format lisible sur iPod ou iPhone ?
- Comment sauvegarder automatiquement un dossier ?
- Comment créer des raccourcis clavier ?

Ne cherchez plus ! Ce livre rassemble les meilleurs astuces pour profiter de votre Mac à 100%

Plus de 400 solutions pour vous dépanner à tout moment : faciles à retrouver, ces techniques indispensables sont classées par grandes catégories thématiques, elle-mêmes présentées par ordre alphabétique, de A comme Airport à W comme Widget.

Les plus de ce livre

- Des solutions testées et approuvées
- Des dizaines d'utilitaires gratuits conseillés
- Des centaines de problèmes courants résolus
- Un index détaillé

Critique du livre par vbrabant

Avoir un accès aisé et faciles à différents trucs et astuces, c'est exactement ce que ce livre permet

Les astuces nous sont clairement exposées, présentées, et la mise en page du livre, le choix de la police, ... en rend sa lecture très agréable

La meilleure façon de lire ce livre est, à mon sens, de choisir un thème dans la table des matières (thème classés par ordre alphabétique), ou de parcourir les entrées de l'index, qui est très bien fourni

Personnellement, je trouve qu'il aurait été plus judicieux de proposer le numéro de l'astuce dans l'index plutôt que le numéro de la page. On aurait, dans certains cas, retrouvé la bonne astuce plus rapidement

Vous aurez très certainement déjà lu ailleurs quelques unes, si pas la majorité des astuces reprises dans ce livre, mais certainement pas toutes. Et puis, là, grâce à ce livre, vous les avez facilement sous la main. Et je suis sûr qu'il y en a, très certainement, que vous ne connaissiez pas.

Certaines astuces mentionnent également des utilitaires, dont certains que je ne connaissais pas du tout et qui se sont avérés très pratiques. Rien que pour ceux là, je suis content de l'achat de ce livre.

En bref, un bon petit recueil d'astuces pour un prix assez modique

Retrouvez ces critiques de livre sur la page livres Mac : [Lien75](#)

Les derniers tutoriels et articles

World Wide Developer Conference 2009 : Le compte rendu de la keynote

La WWDC (World Wide Developer Conférence) 2009 s'est ouverte ce lundi 8 juin 2009 par une keynote qui vous est résumée ici par Marcos Ickx.

1. Introduction

La keynote de la WWDC 09 en a surpris plus d'un.

Tout le monde s'avait qu'on allait parler de Snow Leopard et de iPhone OS 3.0.

Tout le monde s'attendait à ce qu'Apple dévoile un nouvel iPhone.

Mais, personne ne s'attendait à ce qu'Apple annonce de nouveaux MacBook Pro. Et c'est pourtant avec cette annonce qu'Apple a démarré sa keynote.

Mais, si vous le permettez, nous n'allons pas suivre l'ordre de présentation de la keynote.

Nous allons dans un premier temps parler des annonces concernant les Mac (Safari 4, Snow Leopard, la nouvelle gamme des MacBook Pro) Et plus tard, nous aborderons les annonces concernant l'iPhone (iPhone OS 3.0 et iPhone 3G S)

2. Coté Mac

2.1. Snow Leopard

Snow Leopard, la prochaine version de Mac OS X, sera disponible en Septembre 2009. Et ne coutera pas cher du tout si vous possédez déjà Leopard.

Prix de Snow Leopard

- Snow Leopard coutera 29\$ pour les détenteurs

d'une licence Leopard

- Snow Leopard coûtera 49\$ pour les détenteurs d'une licence family pack (5 licences) de Leopard
- Snow Leopard ne coûtera que 9.95\$ pour ceux qui auront acheté un Mac depuis le 8 juin 2009
- Snow Leopard coûtera 129\$ pour les détenteurs d'une version précédente à Leopard.

Snow Leopard reprend les bases de Leopard, mais ils y ont travaillé selon 3 points

- Raffinements
- Technologies
- Microsoft Exchange

2.1.1. Raffinements



Raffiné, de A jusqu'à Z.
Image issue du site web d'Apple

2.1.1.1. Le Finder

Prenons tout d'abord le Finder. Bien que son aspect n'ait pas changé en apparence, il a en fait été réécrit pour supporter les nouvelles technologies introduites dans Snow Leopard.

Ainsi, le finder, comme la majorité des composants livrés avec Snow Leopard, est une application 64 bits et tirera profit de Grand Central Dispatch.

Nous parlerons du 64 bits et de Grand Central Dispatch dans la section "Technologies".

2.1.1.2. Rapidité / Encombrement

L'installation de Mac OS X Snow Leopard sera 45% plus rapide que précédemment, occupera 6Go d'espace de moins que Mac OS X Leopard. Et la sauvegarde initiale pour Time Machine sera plus rapide de 50%.

Le fait que la sauvegarde initiale pour Time Machine soit 50% plus rapide nous paraît tout à fait normal puisque l'espace disque occupé après installation est également réduit de 50%. Donc, Time Machine devra copier 2 fois moins de données lors de la sauvegarde initiale.

Comparé à Leopard, Snow Leopard non seulement s'éteindra plus vite, mais sortira de veille encore plus vite.

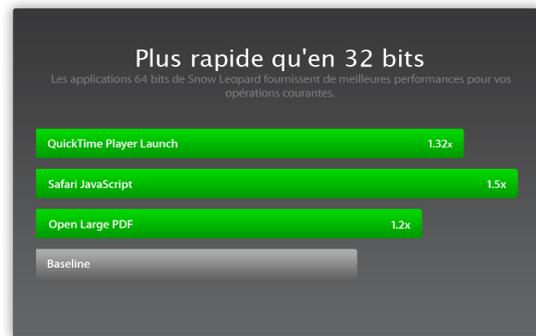
2.1.2. Technologies



2.1.2.1. 64 Bits

Alors que Leopard avait déjà commencé le virage vers le 64 bits, Snow Leopard va encore plus loin puisque la plupart des applications qui sont livrées avec Snow Leopard sont maintenant écrites en 64 bits.

Et cela devrait se voir sur la rapidité d'exécution des tâches. Voici d'ailleurs un aperçu des gains du passage au 64 bits.



2.1.2.2. QuickTime X

La nouvelle version de QuickTime X repose maintenant également sur des bases modernes (comprenez par là, 64 bits, GCD, OpenCL), supportera l'accélération matérielle pour le décodage des vidéos H.264. QuickTime X supportera également le HTTP Streaming, ne nécessitant plus de solution coûteuse à mettre en place côté serveur. Un simple serveur Apache suffira. Et vu que c'est du HTTP, ça passera sans problème les pare-feu.

L'interface visuelle de QuickTime X a également été revue, s'approchant quelque peu de ce qu'on peut voir maintenant lorsqu'on joue des vidéos depuis iTunes.

Aussi, avec QuickTime X, vous pourrez facilement délimiter un morceau de la vidéo que vous êtes en train de regarder pour ensuite l'envoyer par mail, la publier sur youtube, ou MobileMe.

Question vidéo, iChat pourra maintenant gérer les vidéos en 640*480, y compris dans iChat Theater (320*200 auparavant) et mobilisera pour cela moins de bande passante que précédemment.

2.1.2.3. Grand Central Dispatch

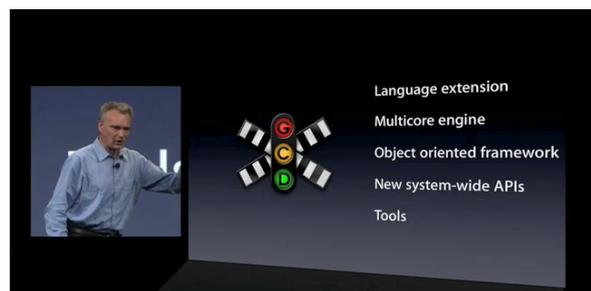


Image extraite de la vidéo Apple de la keynote de la WWDC 09

Cette technologie est promise à un grand avenir vu le nombre de cœurs qui vont continuer à se multiplier dans les années qui viennent.

D'après Apple, Grand Central Dispatch devrait permettre une meilleure répartition des threads vers les différents cœurs du processeur.

Mais, car il y a un mais, Grand Central Dispatch n'est pas juste une technologie qui est fournie avec Snow Leopard,

et qui va jouer, comme par magie le chef d'orchestre.

Non. En fait il faudra que les développeurs d'applications réécrivent leurs programmes pour appeler l'API de Grand Central Dispatch.

A part donc la panoplie des applications livrées avec Snow Leopard (finder, Mail, iChat, Safari, iCal, pour n'en citer que quelques unes), il faudra un certains temps avant que les applications tirent profit de cette technologie.

Mais on peut s'attendre à ce que les prochaines versions des logiciels Apple tirent profit de Grand Central Dispatch.

Apple a mis à disposition de tous un "dossier technique", expliquant en grande ligne le fonctionnement de Grand Central Dispatch. Vous pouvez le retrouver à l'adresse suivante : [\(Lien76\)](#)

2.1.2.4. OpenCL

Apple a réussi un magnifique coup avec "OpenCL". En effet, en moins d'un an, Apple a réussi à faire en sorte qu'OpenCL sorte de l'état de projet interne à l'état d'un projet reconnu par une société de standardisation, disponible en open source, sans royalties, et a réussi à réunir autour de ce projet différents constructeurs de cartes graphiques et des constructeurs du monde la téléphonie mobile.

A noter tout de même un grand absent : Microsoft.

Apple a mis à disposition de tous un "dossier technique", expliquant en grande ligne ce qu'est OpenCL. Vous pouvez le retrouver à l'adresse suivante : [\(Lien77\)](#)

2.1.2.5. Exchange

Apple a décidé de vouloir s'introduire dans le monde des entreprise. Et qui dit monde des entreprises, dit Microsoft Exchange

Apple a donc fait le nécessaire pour que iCal, Mail et le Carnet d'Adresses s'intègrent sans problème avec Exchange.

Faudra même pas acheter, coté client, d'applications supplémentaires. Tout Mac ayant Snow Leopard pourra s'intégrer très facilement avec Exchange.

Mais est-ce que cela suffira pour convaincre les gestionnaires des parcs informatiques pour vous laisser brancher votre Mac sur le réseau de l'entreprise ?

2.2. Les nouveaux MacBook Pro

Apple a profité de la WWDC pour revisiter la gamme des MacBook Pro.

De mon point de vue, la modification la plus importante est le fait que tous les MacBook Pro aient maintenant cette nouvelle batterie qu'Apple avait déjà introduit dans les derniers MacBook Pro 17", et qui vous donne une autonomie incroyable. J'en parlerai d'ailleurs un peu plus en détails dans le point qui suit.

Une des autres modifications, qui va aussi faire couler beaucoup d'encre, c'est le remplacement du port ExpressCard sur le MacBook Pro 15" par un lecteur de carte SD.

Je trouve cela quelque peu chiche. Surtout lorsqu'on sait que l'on retrouve les cartes SD non pas sur les appareils photos destinés aux professionnels, encore moins sur les caméscopes, mais on les retrouve plutôt du côté des appareils grand public.

Le professionnel regrettera lui la disparition de l'Express Card.

Et tous les professionnels de l'image et/ou du son ne sont pas prêt à passer au MacBook Pro 17". Eux aussi tiennent à leur dos, et aimeraient pouvoir continuer à utiliser un MacBook Pro sommes toute moins léger et encombrant qu'un 17".

L'autre changement important au niveau du MacBook Pro 15", est l'apparition de 3 modèles.

Alors que précédemment Apple ne proposait que 2 MacBookPro 15", tous les deux équipés des deux cartes graphiques 9400M et 9600M GT, Apple a rajouté un nouveau 15" d'entrée de gamme, équipé uniquement de la carte graphique 9400M. Ce qui n'est finalement pas une si mauvaise chose que cela car, du coup, cela permet à Apple de proposer le premier MacBook Pro 15" à 1599€ (1649€ pour la Belgique), au lieu de 1799€ auparavant.

Une autre modification annoncée lors de la keynote mais qui a été très peu commentée ensuite sur le net, c'est le fait que les MacBook Pro sont équipé d'un écran de meilleur qualité, ayant un gamut plus large que précédemment, permettant le rendu d'une plus large gamme de couleurs qu'auparavant. Faudrait voir ce qu'un photographe en pense par rapport au précédent.

Et last, but not least. Apple a également décidé de faire rentrer les MacBook 13" Unibody dans la gamme des MacBook Pro.

Ainsi, les MacBook Pro 13" ont maintenant également un écran de meilleur qualité qu'auparavant, un clavier rétro-éclairé, un lecteur de carte SD comme son grand frère, le 15", la batterie à large autonomie intégrée, et, coup de théâtre, le retour du port FireWire 800.



Le MacBook Pro 13 pouces



Le MacBook Pro 15 pouces



Le MacBook Pro 17 pouces

Et pour terminer, voici les nouveaux tarifs :

Le MacBook Pro 13" à 2,26Ghz est au prix de 1149€ et le MacBook Pro 13" à 2,53Ghz est au prix de 1399€.

Le MacBook Pro 15" d'entrée de gamme, qui a exactement les mêmes caractéristiques matérielles que le 13" à 2,53Ghz, excepté la taille de son écran, est à 1599€. Soit 200€ plus cher, pour 2" en plus (et une entrée audio ligne !) Le MacBook Pro 15" de milieu de gamme est vendu à 1799€, et le MacBook Pro 15" de haut de gamme est vendu à 2099€.

Le MacBook Pro 17" voit son prix passer de 2499€ à 2299€.

Le MacBook Air est maintenant disponible à 1399€ et 1699€ pour la version SDD. Une belle baisse de prix.

Mais revenons, si vous le permettez à ce qui me semble le plus gros changement de cette gamme de MacBook Pro 13" et 15" qu'est la batterie.

2.2.1. La batterie

Toute la gamme des MacBook Pro possèdent maintenant la fameuse batterie intégrée d'un nouveau genre qui était apparue pour la première fois sur le MacBook Pro 17" de janvier 2009.

Cette batterie permet, selon Apple, une autonomie de 7 heures (8 pour le MacBook Pro 17") par cycle.

Et cette batterie est faite pour supporter 1000 cycles de charge, contre 300 précédemment.

Ce qui permet à Apple d'affirmer que sa batterie a une durée de vie de 5 ans.

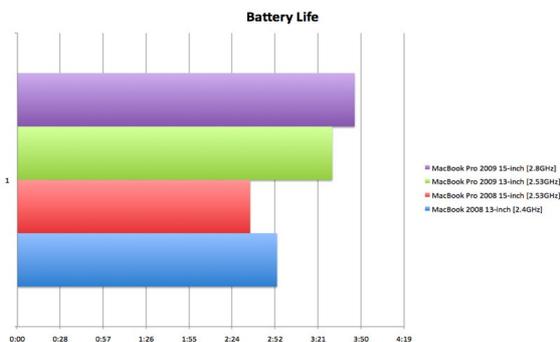
Bien qu'Apple affirme que sa batterie a maintenant une durée de vie de 5 ans (7 heures d'utilisation par jour durant 1000 jours, en comptant 200 jours d'utilisation par an) la garantie de la batterie est toujours d'1 an, pas plus.

Or, ce qui est fabuleux, c'est que d'après les tests effectués par différents sites, tous s'accordent à dire que cette batterie tient vraiment les promesses annoncées par Apple. Et Apple serait même le premier constructeur à donner des chiffres pour sa batterie qui sont en dessous de la réalité.

Petit tour d'horizon :

Gizmodo ([Lien78](#)) annonce lors de son test des nouveaux MacBook Pro de juin 2009, avoir obtenu 3 heures 46 d'autonomie pour le 15" 2.8Ghz et 3heures 31 pour le 13" 2.53Ghz.

Soit un gain d'un peu plus d'une heure, pour le MacBook Pro 15" par rapport à l'ancien modèle.



Résultat des tests d'autonomie réalisé par Gizmodo

Il faut savoir que les conditions de test n'étaient pas les mêmes.

Apple précise sur sa page consacrée à la batterie que les tests se réalisent dans les conditions suivantes

Le test de productivité sans fil mesure l'autonomie de la batterie à travers la consultation sans fil de divers sites web et la modification de texte dans un document de traitement de texte, la luminosité de l'écran étant réglée sur 50 %.

Or Gizmodo a fait les tests dans les conditions suivantes :

Luminosité de l'écran réglée à 50%, rétro-éclairage clavier faible, wifi activé et lecture d'un film codé au format H.264

Ca fait tout de même une grande différence entre surfer et éditeur du texte (ce qui expliquerait les 7 heures avancées par Apple) et regarder un film à l'écran.

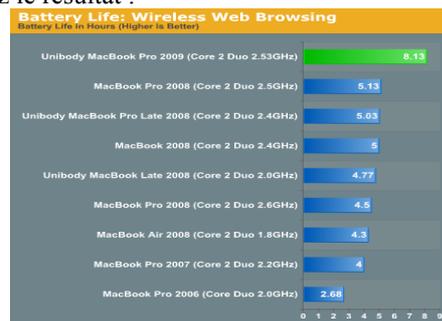
Ce fut ensuite au tour d'AnandTech ([Lien79](#)) de se pencher sur la nouvelle batterie :

Les conditions de test d'AnandTech se rapprochaient assez bien de celles décrites par Apple.

The wireless web browsing test uses the 802.11n connection to browse a series of 20 web pages varying in size, spending 20 seconds on each page (I timed how long it takes me to read a page on Digg and came up with 36 seconds; I standardized on 20 seconds for the test to make things a little more stressful). The test continues to loop all while playing MP3s in iTunes.

Donc, en gros : connexion wifi activée, navigation sur le web, tout en écoutant de la musique MP3 jouée dans iTunes

Et voyez le résultat :



Résultat des tests d'autonomie réalisé par AnandTech

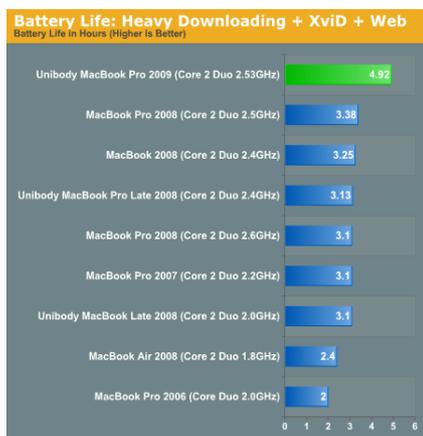
Plus de 8 heures d'autonomie. 8 heures 7 pour être précis. 3 heures de plus que le MacBook Pro 15" de début 2008.

N'en croyant pas ces yeux, il a revu ses tests pour aller surfer sur des pages plus lourdes en images, et surtout qui contenaient maintenant des animations flash, ce qui n'était pas le cas auparavant.

Et là, il obtenait un peu moins de 6heures 30 d'autonomie pour le nouveau MacBook Pro 15".

Continuant plus loin ses tests, il a maintenant rejouer son test précédent tout en téléchargeant un fichier de 10Go sur le net et en jouant des vidéos encodées au format XviD en boucle.

Résultat :

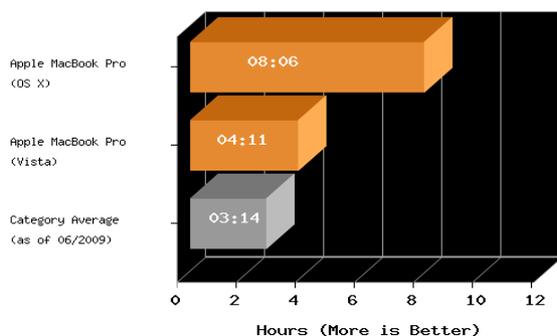


Résultat des tests d'autonomie de la batterie par AnandTech

juste en dessous des 5 heures d'autonomie.

C'est ensuite LaptopMag ([Lien80](#)) qui publie un test d'autonomie des MacBook Pro 15" fort intéressant.

En effet, alors que ce test d'autonomie confirme celui de AnandTech (8heures 6 d'autonomie), il n'est confirmé que sous Mac OS X Leopard.



Résultat des tests d'autonomie du MacBook Pro par LaptopMag

En effet, les mêmes tests effectués sous Vista ne donnent que 4heures 11 d'autonomie.

Puisqu'on parle des tests sous Vista, PCMag ([Lien81](#)) a également testé le MacBook Pro 15" avec les mêmes batteries de test que ceux utilisés pour tester les autres PC.

On peut donc faire une comparaison directe avec d'autres Modèles de PC comme le Dell Studio XPS 16 ou le HP Pavilion HDX16T.

Le MacBook Pro 15" obtient, pour le MobileMark 2007, un résultat de 5heures 30.

Pour avoir une idée de comparaison, le Dell arrive à 4heures 20 d'autonomie, le HP à 2heures 8. l'ancien MacBook Pro 15" arrivait à 2heures d'autonomie.

Mais, comme le dit PCMag, pour que le Dell arrive à 4heures 20 d'autonomie, il a fallu lui mettre une super batterie qui ne tient pas dans la coque du PC.

Cela montre encore une fois combien la gestion de l'énergie est également une affaire logicielle et pas seulement matérielle.

Mais de là à ce que Mac OS X Leopard gère 2 fois mieux l'énergie que Microsoft Windows Vista, je dois avouer que même moi je ne m'y attendais pas.

Avoir aux alentours de 4 heures d'autonomie 3heures 46 si on prend les chiffres de Gizmodo, 5heures si on prend les chiffres d'AnandTech, dans des conditions d'utilisation énergivore c'est tout bonnement génial.

Et avoir plus de 8 heures d'autonomie dans des conditions d'utilisation normale, c'est vraiment tout bonnement incroyable.

Il y a juste un point sur lequel ces batteries doivent encore montrer leurs promesses : supportent-elles vraiment 1000 cycles de recharge ? Tiendront-elles vraiment 5 ans comme annoncé par Apple.

Pour cela, je vous donne rendez-vous dans 5 ans.

Retrouvez la suite de l'article de Marcos Ickx en ligne : [Lien82](#)

OpenSSH : Mise en place et description détaillée d'un serveur SSH

Cet article vise à appréhender l'installation, la configuration et les différentes possibilités d'utilisation d'OpenSSH sous Linux. Il détaille les différentes options de configuration offertes par OpenSSH et les différents types de connexion à un serveur SSH, que ce soit sous un système GNU/Linux, Unix ou Windows.

1. Introduction

OpenSSH ([Lien83](#)) est un ensemble d'outils développés sous licence OpenBSD. Son nom est la contraction d'OpenBSD Secure SHell. OpenSSH permet d'effectuer des communications sécurisées à travers un réseau, reposant sur le protocole SSH. OpenSSH offre une solution de remplacement sécurisé pour rlogin, telnet, rcp et ftp.

OpenSSH est disponible sur un grand nombre de systèmes d'exploitation dont BSD, Linux, AIX, HP-UX, Solaris, Mac OS X, Windows via Cygwin...

SSH est un protocole de communication sécurisée reposant sur le mode client-serveur. SSH signifie *Secure SHell*. SSH offre la confidentialité des échanges et l'authentification des correspondants.

Il existe 2 versions du protocole, la version 2 étant largement utilisée maintenant.

Le protocole SSH utilise par défaut le port 22.

2. Installation et fonctionnement du serveur OpenSSH

2.1. Sur un système Linux

Remarque : La méthode d'installation proposée dans cet article repose sur le fonctionnement de la distribution Debian et ses dérivés. Pour les autres distributions, veuillez vous reporter aux commandes de gestion des paquets implantées pour la distribution choisie.

Installation du serveur OpenSSH :

Installation d'OpenSSH

```
> apt-get install openssh-server
```

L'installation du serveur OpenSSH crée un dossier `/etc/ssh` et génère un couple de clés RSA :

- `ssh_host_rsa_key` contenant la clé privée du serveur
- `ssh_host_rsa_key.pub` contenant la clé publique du serveur

Lorsqu'un client demande une connexion au serveur, ce dernier envoie au client sa clé publique. Le client utilise alors la clé publique reçue pour envoyer au serveur une clé secrète. Le serveur reçoit alors la clé publique et la déchiffre avec sa clé privée, prouvant ainsi qu'il est bien le serveur.

Pour le prouver au client, le serveur crypte un message type avec la clé publique du client et lui envoie. Si le client

arrive à lire le message en le déchiffant avec sa clé privée, il a l'assurance de communiquer avec le véritable serveur. Le canal de communication sécurisé est alors créé.

Si pour une quelconque raison, vous étiez amené à devoir générer à nouveau le couple de clés du serveur, reportez-vous à la partie IV-2-a et nommez les clés `ssh_host_rsa_key`.

2.2. Sous Windows, avec Cygwin

Cette partie ne traite pas de ce qu'est Cygwin et comment l'installer mais de comment installer un serveur SSH sous Cygwin.

Cygwin ([Lien84](#)) est une sorte d'émulateur de système Unix sous Windows. Il est constitué d'un grand nombre de paquets, logiciels libres existants sous Linux dont OpenSSH. Ainsi, il est possible d'avoir un serveur SSH sous Windows !

Pour faire fonctionner un serveur SSH sous Cygwin, il est nécessaire de créer les groupes et utilisateurs de Windows dans Cygwin en copiant ceux-ci de Windows à Cygwin :

```
#Pour les utilisateurs :  
mkpasswd -l > /etc/passwd  
  
#Pour les groupes :  
mkgroup -l > /etc/group
```

Il est nécessaire de donner un mot de passe à votre utilisateur Windows afin de permettre une connexion par SSH !

Les groupes et utilisateurs sont maintenant créés.

Il faut maintenant initialiser la configuration du serveur SSH avec cette commande :

Initialisation du serveur SSH

```
ssh-host-config -y
```

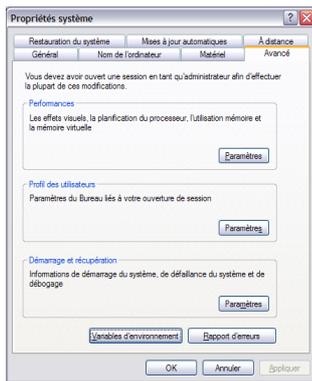
Entrez les informations demandées.

Lorsqu'il vous est demandé "CYGWIN=", la réponse à entrer est : `ntsec tty`.

Comme sous Unix/Linux, Windows dispose de variables d'environnement, notamment le PATH. Il est nécessaire d'éditer le PATH et de rajouter le répertoire des commandes pour Cygwin.

Pour cela, allez dans :

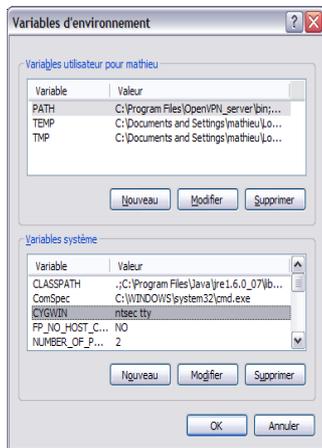
Poste de travail --> Propriétés --> Avancé --> Variables d'environnement



Fenêtre des propriétés système de Windows XP

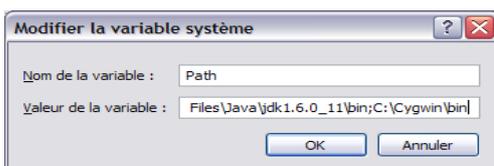
Dans la partie "Variables système", cliquez sur "Nouveau" :

- Le nom de la variable à ajouter est : *CYGWIN*
- La valeur de cette variable est : *nsec tty*



Variable CYGWIN

Ensuite, sélectionnez PATH dans la liste, cliquez sur "Éditer" et ajoutez à la fin du PATH : *C:\Cygwin\bin*.



Variable PATH

Voilà, il ne vous reste plus qu'à autoriser le démarrage du serveur SSH au démarrage de Windows. Pour cela, utilisez les commandes :

```
net start sshd
cygrunsrv -S sshd
```

3. Configuration du serveur d'OpenSSH

La configuration du serveur ssh se résume en un fichier : */etc/ssh/sshd_config*.

La configuration par défaut du serveur est suffisante son fonctionnement.

Voici une description des différents paramètres qui composent le fichier */etc/ssh/sshd_config* :

Certaines des descriptions suivantes sont issues de la traduction de la manpage ([Lien85](#)) de *sshd_config*.

(Le tableau des description est consultable sur l'article en ligne ([Lien86](#)))

D'autres options existent. Vous pouvez les retrouver sur la manpage officielle ([Lien87](#)) de *sshd_config* d'OpenSSH.

Comme souvent, le démarrage, l'arrêt et le redémarrage du serveur se font comme ceci, respectivement :

```
/etc/init.d/ssh start
/etc/init.d/ssh stop
/etc/init.d/ssh restart
```

4. Authentification

Différentes possibilités existent concernant l'authentification pour la connexion au serveur SSH : l'authentification par mot de passe et l'authentification par échange de clés. Voici comment mettre en place ces deux techniques.

4.1. Par login/mot de passe

Cette méthode d'authentification est très simple, il suffit d'exécuter la commande de connexion SSH :

```
ssh login@nom_du_serveur
# ou
ssh login@IP_du_serveur
```

Lors de la première connexion SSH au serveur avec ce login, il vous est demandé si le fingerprint de la clé publique du serveur est bon. Si le fingerprint de la clé publique du serveur est bien le même que celui affiché à votre écran, vous pouvez accepter de continuer en tapant *yes*.

La clé publique du serveur SSH est alors copiée dans le fichier *~/.ssh/known_hosts*, ceci permettant garder en mémoire l'authenticité de la clé publique du serveur pour les prochaines connexions.

Ensuite, le mot de passe de l'utilisateur utilisé pour se connecter vous est demandé.

Première connexion par SSH - Fingerprint

```
The authenticity of host '[192.168.0.7]
([192.168.0.7])' can't be established.
RSA key fingerprint is
12:12:m3:df:6f:23:3b:18:99:ea:84:p6:d1:de:f6:71.
Are you sure you want to continue connecting
(yes/no)? yes
Warning: Permanently added '[192.168.0.7]' (RSA)
to the list of known hosts.
root@192.168.0.7 s password:
```

4.2. Par échange de clés

Cette méthode d'authentification repose sur la cryptographie asymétrique ([Lien88](#)), par l'utilisation du couple clé privée/clé publique créé par l'utilisateur client, de la même façon que le fait le serveur.

4.2.1. Génération des clés

SSH propose de générer le couple de clés publique/privée

en utilisant au choix 2 algorithmes : DSA et RSA.

Utiliser au choix RSA ou DSA.

Pour ce faire, OpenSSH propose un outil de génération très simple à utiliser : ssh-keygen.

Génération du couple clé privée / clé publique pour le user client

```
ssh-keygen -t rsa
```

A l'exécution de cette commande, le couple de clés est généré. Il vous est alors demandé le chemin où vous voulez stocker ces clés et leur nom. La touche "entrée" permet de valider le chemin proposé par défaut. La clé privée prend les droits 600 et la clé publique 644. Suite à cela, il vous est demandé d'entrer une passphrase qui vous permettra de protéger votre clé privée. Cette passphrase peut contenir tout caractère et "méta-caractère", même des espaces. Elle est à retenir car elle vous est demandée à chaque connexion par échange de clés.

```
user@debian:~/ssh$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key
(/home/user/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in
/home/user/.ssh/id_rsa.
Your public key has been saved in
/home/user/.ssh/id_rsa.pub.
The key fingerprint is:
f1:68:21:f2:b5:61:c2:cd:22:c4:48:e8:ab:c2:12:z7
user@debian
The key's randomart image is:
+--[ RSA 2048 ]-----+
|      o++o      |
|     o..o.o o   |
|    . o = + .   |
|   . . o +     |
|  . P . S .    |
| . =   o o     |
|.. .      . .  |
|o..        .   |
|.o         |
+-----+
```

4.2.2. Diffusion de la clé publique

Pour permettre l'authentification par échange de clés, il est nécessaire de donner votre clé publique au serveur, selon de principe de la cryptographie asymétrique.

Encore une fois, OpenSSH à penser à tout et vous offre un outil pour copier votre clé publique au serveur en toute sécurité : ssh-copy-id

```
ssh-copy-id -i chemin/clé/publique/id_rsa.pub
login@serveur_ssh
```

A cet instant, le fichier known_hosts est alors lu pour voir si la machine est connue. Puis, le mot de passe session de l'utilisateur "login" est demandé et ce sera la dernière fois. Les prochaines connexions avec cet utilisateur se feront donc par échange de clé privée/clé publique et à l'aide de la passphrase entrée lors de la génération des clés de l'utilisateur.

Bien que ssh-copy-id soit conçue spécialement pour copier votre clé publique sur le serveur SSH, il est possible d'effectuer cette tâche d'une autre façon, avec scp :

Copie de la clé publique cliente sur le serveur SSH avec scp

```
scp /home/login/.ssh/id_rsa.pub
login@serveur_SSH:/home/login/clé_publicque

ssh login@serveur_SSH
cat clé_publicque >>
/home/login/.ssh/authorized_keys
rm clé_publicque
```

4.2.3. Connexion ssh

De cette manière donc, la connexion ssh ne se fait plus par login/mot de passe mais grâce à l'échange de clés privées/clés publiques et à l'utilisation de la passphrase.

La commande de connexion reste cependant la même :

```
ssh login@nom_du_serveur
# ou
ssh login@IP_du_serveur
```

4.3. Au revoir mots de passe, passphrase...

Si vous utilisez ssh régulièrement, vous avez du vous rendre compte que cela devient très vite fatigant de devoir retaper à chaque connexion ssh son mot de passe ou sa passphrase.

Il existe 2 alternatives à ce désagrément : l'authentification sans mot de passe et ssh-agent.

4.3.1. L'authentification sans mot de passe

L'authentification sans mot de passe est très pratique et pratiquement incontournable pour exécuter des scripts distants, par exemple un script shell d'arrêt global se connectant à tour de rôle sur vso différents serveurs pour exécuter la commande spécifique d'arrêt.

L'authentification sans mot de passe est donc très efficace mais pose cependant un problème de sécurité non négligeable.

Son principe est tout simple :

Cela consiste en la génération d'un couple de clé privée/clé publique, de la même manière que pour l'authentification par échange de clés et en la diffusion de votre clé publique. Cependant, au moment de rentrer une passphrase comme il l'est demandé au cours de la génération du couple de clés, vous n'entrez rien et vous faites "Entrée". Répétez cette touche à la demande de re-saisie de la passphrase.

Cela a pour effet de renseigner la chaîne vide comme passphrase, ce qui revient à dire qu'il n'y a pas de passphrase.

Ainsi, lors de la connexion par ssh, l'étape de demande de passphrase est "omise" et la connexion s'effectue sans demande "de vérification au clavier".

Authentification sans mot de passe

```
debian:/home/user# ssh login@IP_serveur
Linux lenny.debian.com 2.6.26-1-686 #1 SMP Fri
Mar 6 11:08:15 UTC 2009 i686

The programs included with the Debian GNU/Linux
```

```
system are free software;
the exact distribution terms for each program are
described in the
individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO
WARRANTY, to the extent
permitted by applicable law.
lenny:~#
```

4.3.2. Utilisation de ssh-agent

Ssh-agent est un autre outil développé par OpenSSH permettant d'utiliser l'authentification par échange de clés avec une passphrase sans avoir à toujours taper cette dernière.

Une fois ssh-agent démarré, toutes les commandes ou programmes exécutés par ssh sont des sous-processus de ssh-agent, ne nécessitant plus d'authentification ultérieure.

Ainsi, il ne reste donc plus qu'à récupérer les variables d'environnement, de les mettre en place pour ssh-agent et demander à ssh-agent de s'occuper de notre clé publique et le tour est joué...

récupération des variables d'environnement

```
ssh-agent
```

Les variables d'environnement s'affiche à l'écran. Pour les mettre en place pour ssh-agent, il suffit de copier les 3 lignes de l'écran et de les recoller dans la console.

Mise en place des variables d'environnement pour ssh-agent

```
user@debian:~$ ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-TubtX25143/agent.25943;
export SSH_AUTH_SOCK;
SSH_AGENT_PID=25984; export SSH_AGENT_PID;
echo Agent pid 25984;
```

ssh-add

```
ssh-add ~/.ssh/id_dsa.pub
# ou
ssh-add id_rsa.pub
```

C'est fini... Enfin presque. En effet, ssh-agent n'est "valable" que dans la console dans laquelle il a été lancé. Ce qui signifie que si vous fermez votre console et en ouvrez une autre, il faut tout recommencer.

4.3.3. Automatiser l'utilisation de ssh-agent

Comme il a été dit à la fin de la partie IV-3-b, ssh-agent n'est valable que pour la console dans laquelle il est lancé. Ainsi, pour optimiser l'utilisation de ssh-agent et éviter de devoir recommencer le lancement de ssh-agent et la copie des variables d'environnement à chaque fois, des personnes se sont penchés sur le problème (notamment Joseph M. Reagle ([Lien89](#))) et on écrit un bout de code pour pallier ce problème.

Ce code est à coller dans le fichier `.x_profile` de votre HOME. Par exemple, avec un shell en Bash, le fichier en question est : `/home/votre_user/.bash_profile`

Code permettant de ne plus avoir à lancer ssh-agent à chaque nouvelle console. Ce code est à placer dans le

.x_profile de votre home

```
SSH_ENV="$HOME/.ssh/environment"

function start_agent {
    echo "Initialising new SSH agent..."
    /usr/bin/ssh-agent | sed 's/^echo/#echo/' >
"${SSH_ENV}"
    echo succeeded
    chmod 600 "${SSH_ENV}"
    . "${SSH_ENV}" > /dev/null
    /usr/bin/ssh-add;
}

# Source SSH settings, if applicable

if [ -f "${SSH_ENV}" ]; then
    . "${SSH_ENV}" > /dev/null
    ps -ef | grep ${SSH_AGENT_PID} | grep ssh-
agent$ > /dev/null || {
        start_agent;
    }
else
    start_agent;
fi
```

Ainsi, avec ce code, il n'est plus nécessaire de lancer ssh-agent. Vous n'avez qu'à copier votre clé sur le serveur si ce n'est pas déjà fait et vous connecter en ssh sans besoin de mot de passe.

5. Un peu de sécurité

5.1. Changer le port d'écoute côté serveur

Comme dit plus haut, le port d'écoute par défaut d'un serveur SSH est le port 22. Or, vous remarquerez que très souvent, des machines inconnues tentent de se connecter par SSH sur vos serveurs grâce à des scripts automatisés testant sur le port 22 un grand nombre d'utilisateurs et mot de passe couramment utilisés tels que root, mysqladmin, oracle...

Une première barrière pour éviter toute intrusion par SSH est de changer le port d'écoute du serveur SSH.

Les ports jusqu'à 1024 sont "réservés" pour diverses protocoles et applications. Ainsi, choisissez un port de nombre supérieur à 1024.

Pour changer le port d'écoute, il suffit d'éditer le fichier de configuration côté serveur : `/etc/ssh/sshd_config` :

Changement du port d'écoute du serveur SSH

```
vi /etc/ssh/sshd_config

#la ligne à éidter est :
Port 22

#Changer le 22 en un autre nombre
```

Lorsque le port de connexion au serveur SSH est changé, différent du port par défaut 22, il est nécessaire de le renseigner dans la ligne de commande de connexion en utilisant l'option `-p`.

Par exemple, si vous avez modifier le port par défaut et choisi d'utiliser le port 2222, la ligne de commande de connexion devient :

```
ssh -p 2222 login@serveur_SSH
```

5.2. Bloquer l'accès à Root par SSH

Les scripts automatisés (notamment) qui tentent de se connecter par SSH sur vos serveurs utilisent principalement les logins couramment utilisés dont Root fait bien sûr partie.

Le fait de laisser l'accès à Root par SSH fait gagner une étape à un pirate qui souhaite s'introduire dans votre système. En effet, le fait de bloquer l'accès à Root entraîne la nécessité à un pirate de trouver un nom d'utilisateur autre, existant sur votre système. Sans cela, le pirate dispose donc du nom d'utilisateur, il ne lui reste plus qu'à trouver le mot de passe de Root pour s'introduire.

Cette fois encore, tout se passe dans le fichier de configuration du serveur SSH `/etc/ssh/sshd_config`. Une option permet de bloquer l'accès à Root, qu'il faut placer à "no" :

```
Option de /etc/ssh/sshd_config permettant de bloquer l'accès à Root par SSH
PermitRootLogin no
```

5.3. Restreindre l'accès à des utilisateurs et/ou groupes spécifiques

La section précédente montre comment bloquer l'accès SSH au super-utilisateur Root. Cependant, par défaut, SSH autorise l'accès à tous les utilisateurs de tous les groupes.

De nombreux logiciels créent automatiquement un utilisateur et/ou un groupe spécifique pour les administrer. Ces utilisateurs possèdent un mot de passe par défaut qui, s'il n'est pas modifié, peut permettre à un pirate d'accéder facilement à votre machine en utilisant ce couple de login/mot de passe, tel que oracle/oracle.

Il est donc possible et fortement conseillé de restreindre l'accès au serveur SSH à quelque(s) utilisateur(s) privilégié(s) de confiance. Plusieurs options sont disponibles pour effectuer cette restriction, au niveau du fichier de configuration du serveur SSH `/etc/ssh/sshd_config` :

Restriction d'accès SSH par autorisation limitée

```
AllowUsers Foo Bar
AllowGroups SSH_serv
```

Cet exemple utilise les options "Allow...". Ces deux options permettent de définir les utilisateurs et groupes autorisés à se connecter par SSH. Seuls les utilisateurs *Foo* et *Bar*, ainsi que les utilisateurs appartenant au groupe *SSH_serv* peuvent donc se connecter. Tous les autres voient leur accès par SSH bloqué.

Il existe une autre façon de restreindre l'accès :

Restriction d'accès SSH par blocage

```
DenyUsers Foo Bar
DenyGroups SSH_serv
```

Dans ce nouvel exemple, la restriction ne se fait plus par la définition d'une liste d'utilisateurs et groupes autorisés à se connecter mais par la définition des utilisateurs et groupes n'ayant pas le droit de se connecter.

Ici, tous les utilisateurs peuvent se connecter par SSH à l'exception des utilisateurs *Foo* et *Bar*, ainsi que les

membres du groupe *SSH_serv*.

6. Transfert de fichiers sécurisé : SCP

6.1. Fonctionnement de la commande scp

Il existe de nombreuses façons de transférer des fichiers d'une machine à une autre, ftp, rcp... mais celles-ci ne sont pas sécurisées. OpenSSH propose donc une commande permettant ce transfert à travers un canal chiffré, reposant sur le protocole SSH : scp.

Son utilisation est très simple. C'est une combinaison de cp et de ssh.

Dans le premier exemple, le fichier `/var/log/data.txt` est transféré de la machine locale vers le HOME de l'utilisateur SSH utilisé pour la connexion au serveur SSH :

```
scp /var/log/data.txt
login@serveur_SSH:/home/login/
```

Dans ce second exemple, le fichier `/var/log/data.txt` est transféré du serveur SSH vers le HOME de l'utilisateur local "moi" :

```
scp login@serveur_SSH:/var/log/data.txt
/home/moi/
```

6.2. Scp par un port différent du port 22

Dans le paragraphe V qui traite de la sécurité, il est conseillé d'utiliser un port différent du port 22 par défaut d'écoute du serveur SSH. Or, comme il a été vu plus haut, cette action entraîne le fait de devoir spécifier le numéro du port utilisé lors de la connexion par SSH.

Scp utilisant SSH, il en est donc de même pour lui, il est nécessaire de renseigner le port à utiliser pour contacter le serveur SSH dans la commande. Cela se fait avec l'option `-P`.

la commande ssh utilise l'option `-p` ("p" minuscule) pour spécifier le numéro de port à utiliser, alors que scp utilise l'option `-P` (P majuscule)

scp vers un serveur SSH écoutant le port 2222

```
scp -P 2222 /var/log/data.txt login@serveur_SSH:/
home/login/
```

7. Utiliser une application graphique à distance : X11 Forwarding

OpenSSH permet de se connecter de manière sécurisée sur un ordinateur distant. Une des diverses options que propose OpenSSH est le X11 forwarding. Cela consiste à faire du déport d'affichage sur une machine cliente et ce, à travers un canal chiffré.

Par exemple, si vous souhaitez travailler chez vous sur une application graphique coûteuse, installée uniquement sur le serveur de votre entreprise, cela est possible de le faire en toute sécurité.

Les prérequis sont :

- disposer d'une connexion SSH avec le serveur.
- disposer d'un serveur graphique sur la machine cliente

- un débit suffisant pour faire transiter les informations graphiques.

Pour le X11 Forwarding, il n'est pas nécessaire d'avoir d'installé un serveur graphique sur le serveur, mais juste sur la machine devant afficher la fenêtre graphique de l'application.

Comme souvent, pour activer l'option X11 Forwarding sur le serveur SSH, il suffit de mettre à "yes" l'option *X11Forwarding* dans le fichier de configuration du serveur SSH : `/etc/ssh/sshd_config` :

Option X11Forwarding

```
X11Forwarding yes
```

Pour ouvrir une session SSH sur le serveur en autorisant le déport d'affichage, il faut rajouter l'option `-X` à la commande de connexion :

```
ssh -X login@serveur_SSH
```

De plus amples informations sur le X11 Forwarding sont disponibles dans l'article de Davidbrzc et ovh : Exécution d'applications X à distance ([Lien90](#))

8. Utiliser SSH en tant que tunnel sécurisé : TCP Forwarding

Le TCP Forwarding, ou tunnel SSH, permet d'accéder à distance à des services théoriquement inaccessibles depuis l'extérieur du réseau de façon sécurisée. Par exemple, vous pouvez grâce à ce procédé accéder à l'intranet de votre entreprise depuis l'extérieur, vous connecter à votre serveur mail ou bien encore traverser un pare-feu.

Pour être plus clair, voici un schéma théorique de ce qui se passe lors d'une communication par un tunnel SSH :

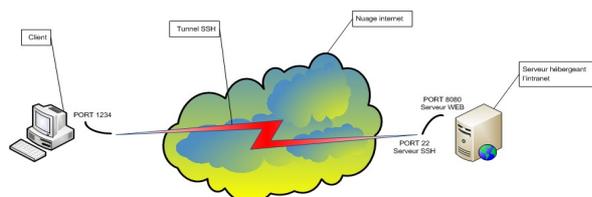


Schéma de principe d'un tunnel SSH

Ce schéma montre la communication pour accéder à l'intranet de votre entreprise depuis l'extérieur.

- Le serveur Web hébergeant l'intranet de votre entreprise utilise le port 8080.
- 1234 est le port côté client depuis lequel le tunnel SSH est créé (remarque : utilisez un port supérieur à 1024).
- Pour des raisons de sécurité, le port du serveur SSH a été changé, il est accessible sur le port 2222.

Ainsi pour créer le tunnel SSH, la commande est :

Etablissement d'un tunnel SSH vers le serveur web de l'intranet

```
ssh -p 2222 -N -L 1234:serveur_web_SSH:8080 login@serveur_web_SSH
```

Voilà, le tunnel est créé. Maintenant pour accéder à l'intranet, il suffit de changer les paramètres de votre navigateur web préféré et de renseigner l'utilisation du proxy : *localhost* par le port : *1234*.



Fenêtre des paramètres de connexion de votre navigateur

9. Conclusion

Avec cet article, j'ai essayé de vous présenter les nombreuses possibilités d'utilisation de SSH dont l'utilité n'est plus à prouver.

Retrouvez l'article de Mathieu Androz en ligne : [Lien86](#)

Bases de données relationnelles et normalisation

Suite aux discussions auxquelles nous avons participé sur les forums Developpez.com concernés, nous avons tiré un certain nombre d'enseignements quant aux questions que se posent certains qui découvrent la normalisation, ou d'autres qui ne l'ont pas abordée par le bon côté. Nous tentons ici de répondre au mieux à leurs attentes tout en formalisant un peu plus et en les creusant, certains points en général juste effleurés.

Bonne lecture.

1. Première forme normale

Beaucoup de choses vraies ou fausses ont été dites au sujet de la première forme normale (1NF), aussi un retour aux sources ne sera-t-il pas de trop. Rappelons une fois de plus que c'est Ted Codd qui a inventé et théorisé le concept de normalisation pour les bases de données, et il serait malvenu de chahuter sans raison profonde les définitions qu'il en a données. Les théoriciens du relationnel sont restés en phase avec ce qu'a écrit Codd, tout en fournissant à l'occasion quelques précisions. Mais le Modèle Relationnel de Données n'est pas figé pour l'éternité et avec le temps il a connu et connaîtra encore des évolutions. Ainsi, sans se départir de l'esprit insufflé par son inventeur, Date et Darwen ont-ils été amenés une vingtaine d'années plus tard à approfondir la 1NF, et ce avec une extrême rigueur.

1.1. La situation en 1969

Dans son tout premier article concernant le Modèle Relationnel de Données (cf. [Codd 1969] page 2), Codd fournissait un exemple (ici francisé) de représentation plate d'une relation de degré 4, décrivant les livraisons de pièces par des fournisseurs, à l'usage de projets, selon certaines quantités :

Livraison (Fournisseur Pièce Projet Quantité)				
	1	2	5	17
	1	3	5	23
	2	3	7	9
	2	7	5	4
	4	1	1	12

Figure 2.1 - Les livraisons, représentation traditionnelle

Cette représentation tout à fait classique et « sage » n'appelle pas de commentaires particuliers.

Dans ce même article, Ted Codd avait écrit (page 3) :

« Nous avons traité de relations définies sur des domaines **simples** — domaines dont les éléments sont des valeurs **atomiques** (non décomposables). Les valeurs non atomiques peuvent être prises en considération dans le cadre relationnel. Ainsi, certains domaines peuvent avoir des relations comme éléments. Ces relations peuvent à leur tour être définies sur des domaines non simples, et ainsi de

suite.

[...] L'adoption d'une perception relationnelle des données, autorise le développement d'un sous-langage universel de recherche, basé sur le **calcul des prédicats du deuxième ordre**.

[...] Le calcul des prédicats du deuxième ordre est nécessaire (plutôt que celui du premier ordre) parce que les domaines sur lesquels les relations sont définies peuvent à leur tour contenir des éléments qui sont des relations. »

Et reprenant l'exemple précédent, Codd propose une représentation sous forme de relations emboîtées :

```
P (Fournisseur, Q (Pièce, R (Projet, Quantité)))
```

Figure 2.2 - Les livraisons, avec emboîtement de relations

Cette fois-ci, la relation P est binaire (degré 2), ses domaines sont Fournisseur (simple) et Q (non simple). La relation emboîtée Q est à son tour binaire et ses domaines sont Pièce (simple) et R (non simple). Les domaines de la relation emboîtée R sont tous simples.

1.2. 1970 : Acte de naissance de la première forme normale

En 1970, dans son article de loin le plus cité et considéré comme celui dans lequel sont posées les fondations du Modèle Relationnel de Données [Codd 1970], Codd préconise l'utilisation exclusive de domaines simples (page 381). A partir de là, *exit* le problème (s'il existe) du calcul des prédicats du deuxième ordre adopté l'année précédente :

« L'utilisation d'un modèle relationnel de données [...] permet de développer un sous-langage universel basé sur le calcul des prédicats. Si la collection des relations est en **forme normale**, alors un **calcul des prédicats du premier ordre** est suffisant. »

« Forme normale » étant à interpréter aujourd'hui comme « Première forme normale » (1NF). Le concepteur de

bases de données relationnelles — qu'il en soit conscient ou non — applique la 1NF à la lettre parce que les domaines des attributs de ses tables sont simples. Même chose pour celui qui conçoit des MCD au sens Merise et qui applique la règle dite de vérification, selon laquelle : « Dans chaque occurrence d'individu-type ou de relation-type on ne trouve qu'une seule valeur de chaque propriété » [TRC 1989].

A noter que divers chercheurs et auteurs (par exemple [Jaeschke 1982], [Abiteboul 1984], [Korth 1988]) ont fait des propositions pour « étendre » le Modèle relationnel et réintroduire la possibilité d'emboîter les relations les unes dans les autres. Les systèmes dans lesquels on s'affranchit de la première forme normale sont dits « Non première forme normale » (*Non First Normal Form*, NF² en abrégé). Il s'agit d'un sujet qui sort de notre périmètre et que nous ne traiterons pas, d'autant plus que Date et Darwen ont montré que l'emboîtement des relations (en tant que valeurs) était possible sans avoir à étendre le Modèle relationnel (cf. paragraphe 2.6).

1971 : Définition de la première forme normale

Dans son article de 1970, Codd décrit la 1re forme normale et c'est en 1971 qu'il en donne la définition (cf. [Codd 1971], page 31) :

« Une relation est en première forme normale si aucun de ses domaines ne peut contenir des éléments qui soient eux-mêmes des ensembles. Une relation qui n'est pas en première forme normale est dite non normalisée. »

Autrement dit, une relation (ceci vaut pour une table SQL) est en première forme normale (1NF) si aucun de ses attributs ne peut prendre de valeur qui soit elle-même un ensemble, en l'occurrence une relation, contrairement donc à ce qu'il avait écrit l'année précédente.

Selon les critères de Codd, si un SGBD acceptait l'instruction suivante, alors il autoriserait pertinemment le viol de la 1NF. En effet, l'attribut Messages est propre à contenir des relations (disons des tables dans un contexte SQL) :

```
CREATE TABLE Membre
(
  MbrId          Integer          Not Null
  , Pseudonyme   VarChar(16)     Not Null
  , DateInscription TimeStamp    Not Null
  , Localisation  VarChar(48)     Not Null
  , AdrCourriel  VarChar(128)    Not Null
  , Messages (    MessageId       Integer          Not Null
                 , MessageDate   TimeStamp        Not Null
                 , MessageTexte  Varchar(Max)     Not Null
                 , Forum          Char(8)         Not Null)
  , Constraint MbrPk Primary Key (MbrId)
  , Constraint CK2 Unique (Pseudonyme)
  , Constraint CK3 Unique (AdrCourriel) ;
```

Figure 2.3 - Viol de la 1NF

Pour rester en accord avec Codd, la structure précédente peut être transformée exactement comme il le montre dans son article de 1970, en procédant par scissiparité, ce qui donne lieu à deux structures, Membre et Message :

```
CREATE TABLE Membre
(
  MbrId          Integer          Not Null
  , Pseudonyme   VarChar(16)     Not Null
  , DateInscription Date         Not Null
  , Localisation  VarChar(48)     Not Null
  , AdrCourriel  VarChar(128)    Not Null
  , Constraint MbrPk Primary Key (MbrId)
  , Constraint CK2 Unique (Pseudonyme)
  , Constraint CK3 Unique (AdrCourriel) ;

CREATE TABLE Message
(
  MbrId          Integer          Not Null
  , MessageId     Integer         Not Null
  , MessageDate   TimeStamp       Not Null
  , MessageTexte  Varchar(Max)    Not Null
  , Forum         Char(8)         Not Null
  , Constraint MessPk Primary Key (MbrId, MessageId)
  , Constraint MessAk Foreign Key (MbrId)
                                     References Membre (MbrId) On Delete Cascade ;
```

Figure 2.4 - Respect de la 1NF

(La contrainte MessAk n'est présente que pour des raisons de cohérence entre les deux tables).

1.3. Années 1970-1980. Les théoriciens du Modèle relationnel sont-ils en phase avec Codd ?

On a brièvement noté que des auteurs comme Roth, Korth et Silberschatz proposèrent une extension du Modèle relationnel afin de réintroduire le principe des relations emboîtées les unes dans les autres (relations NF²).

Mais de façon générale, les théoriciens du relationnel ont fourni dans leurs écrits des définitions de la 1NF conformes à celle de Codd, y compris du reste ceux qui viennent d'être cités. Par exemple :

— « Un schéma de relation R est en première forme normale (1NF) si les domaines de l'ensemble des attributs de R sont atomiques. Un domaine est atomique si ses éléments sont indivisibles. » [Korth 1986]

— « Une relation R est en première forme normale (1NF) si et seulement si tous les domaines sous-jacents contiennent uniquement des valeurs atomiques. » [Date 1986] - Plus tard, Date modifiera sa définition, cf. paragraphe 2.7.

— « Une relation est en première forme normale si tout attribut contient une valeur atomique. » [Gardarin 1988]

— « Une relation est dite "normalisée" ou en "première forme normale" (1NF) si aucun attribut qui la compose n'est lui-même une relation, c'est-à-dire, si tout attribut est atomique (non décomposable). » [Miranda 1986]

Codd lui-même reprit le concept d'atomicité et écrivit en 1990, dans son ultime ouvrage de référence traitant du Modèle Relationnel de Données [Codd 1990] :

« Les valeurs des domaines sur lesquels chaque relation est définie doivent être **atomiques** [...] »

D'autres considèrent, à juste titre, que les relations sont *de facto* normalisées 1NF parce qu'ils ne traitent par définition que de domaines dont les valeurs ne sont pas des ensembles (Delobel et Adiba, Ullman, ...)

Par exemple, Jef Ullman se contente d'écrire (cf. [Ullman 1982], « Normal forms for relation schemes ») :

« Nous n'utilisons pas de domaines dont les valeurs sont des ensembles, et nous pouvons donc nous dispenser de mentionner la première forme normale. Nous considérons en effet que "relation" est synonyme de "relation en première forme normale." »

1.4. L'esprit et la lettre

La définition de la 1NF par Ted Codd est précise. Mais on se situe parfois à la limite quant à son interprétation par les utilisateurs. Par exemple, le catalogue de DB2 for z/OS comporte une table appelée SYSCOPY, utilisée pour les opérations de *recovery*, laquelle comporte une colonne DSVOLSER de type VARCHAR(1784), utilisée pour stocker la liste des numéros à 6 chiffres (séparés par des virgules) des disques hébergeant les données d'une table. L'instruction SQL suivante est conforme du point de vue de la normalisation :

```
INSERT INTO SYSCOPY (... , DSVOLSER, ...)
VALUES (... , "000701,010002,...,314116", ...)
```

En effet, au sens de Codd, la chaîne de caractères "000701,010002,...,314116" est **atomique**, car c'est une valeur tirée du domaine VARCHAR(1784) lequel, à la lettre, est bien atomique, en vertu de quoi la table SYSCOPY respecte la 1NF. Évidemment, si on veut savoir quelles tables (en fait quels *table spaces*) sont hébergées par le disque "314116", il suffit en SQL de coder par exemple :

```
SELECT DISTINCT TSNAME
FROM SYSCOPY
WHERE DSVOLSER LIKE ("%314116%");
```

Et pourtant, on pourrait discuter sans fin du bien-fondé de l'organisation de cette liste de numéros. En tout cas, on peut examiner cet exemple à la lumière de ce qu'a écrit Serge Miranda [Miranda 1988] :

« Une relation est dite normalisée (ou en "première forme normale" notée 1NF) si chaque valeur d'attribut est atomique (c'est-à-dire n'est pas un ensemble ou une liste) ; tout attribut d'une relation 1NF doit donc être monovalué. »

Ou du jugement sans appel de Chris Date ([Date 2007a], page 129) :

« Une colonne (attribut) C constitue un "groupe répétitif" si étant définie sur le domaine D, les valeurs légales pouvant apparaître dans C sont des ensembles (ou des listes, des tableaux, ou ...) de valeurs du domaine D. Ces groupes répétitifs sont définitivement proscrits du Modèle relationnel. »

Ne nous méprenons pas, un attribut peut faire référence à un domaine de chaînes de caractères, de polygones, de tableaux, de relations, de ce que l'on veut. Ce qui est proscrit ici est bien le caractère **répétitif** des valeurs au sein d'un attribut faisant référence à un domaine (type) donné.

En tout état de cause, la table SYSCOPY ne contrevient pas à la 1NF. Une façon d'éviter toute équivoque consisterait à y contrevenir sciemment, comme le permet

PostgreSQL, grâce aux tableaux de valeurs. Au lieu d'écrire :

```
CREATE TABLE SYSCOPY
(DBNAME CHAR(8) NOT NULL,
TSNAME CHAR(8) NOT NULL,
...
DSVOLSER VARCHAR(1784) NOT NULL,
...);
```

Avec PostgreSQL il suffirait d'écrire, en appelant un chat un chat :

```
CREATE TABLE SYSCOPY
(DBNAME CHAR(8) NOT NULL,
TSNAME CHAR(8) NOT NULL,
...
DSVOLSER TEXT[] NOT NULL,
...);
```

Puis :

```
INSERT INTO SYSCOPY
VALUES ("fsmrelb1", "fsmrelt1", ..., ARRAY
["000701", "010002", ..., "314116"], ...)
```

Il est quand même écrit dans la documentation de PostgreSQL :

« Chercher des éléments particuliers dans un tableau peut être le signe d'une **mauvaise conception** de la base de données. Étudiez la mise en place d'une table séparée dont chaque ligne soit affectée à chaque élément du tableau. La recherche sera plus simple et cela sera plus adapté si ces éléments sont nombreux. »

Qui plus est, l'affectation d'une **séquence** à des éléments contrevient à l'esprit du Modèle relationnel.

1.5. L'atomicité : un critère absolu ?

Revenons sur les définitions ci-dessus de la 1NF et dans lesquelles l'accent est donc mis sur l'atomicité. Comme l'a écrit Codd, *Atomique* signifie non décomposable, **pour autant que le système est concerné** : si une donnée réputée atomique cache en réalité une structure complexe, le système n'en a pas connaissance. Il en va ainsi de la colonne DSVOLSER de table SYSCOPY, dans laquelle se cache une liste comme nous l'avons observé précédemment.

Examinons maintenant le contenu de la table Membre de la Figure 2.4, réputée en 1NF :

MbrId	Pseudonyme	DateInscription	Localisation	AdrCourriel
1	Mezig	01/04/2003	Pantin	mez@machin.fr
2	Cezig	02/04/2003	Alpha Centauri	cez@alphacent.zz
3	Tezig	02/04/2003	Tahiti	tez@machin.pf
...				

Figure 2.5 - Valeurs atomiques

À l'intersection de chaque ligne et de chaque colonne, on a exactement une valeur et celle-ci est **atomique**, non décomposable en l'état par le SGBD. Ainsi, la date d'inscription d'un membre est définie comme prenant ses valeurs dans le domaine DATE : ce domaine fait partie des

domaines de base proposés par le système, et si ce dernier n'a pas *a priori* la connaissance sémantique de ce que à quoi on a accès au niveau subatomique, à savoir le jour, le mois, l'année, il met néanmoins à notre disposition des fonctions *ad hoc* correspondantes, telles DAY, MONTH, YEAR. De même, si l'on définissait une colonne CodePostal de type CHAR(5) pour les codes postaux des communes, le système n'aurait pas la connaissance sémantique de la structure interne de ces codes, à savoir par exemple que les deux premiers caractères représentent le numéro du département du bureau distributeur du courrier de la commune (les trois premiers caractères en France d'outre-mer). Cependant, le système nous permet, là encore, d'accéder au niveau subatomique, au moyen de la fonction SUBSTRING ou du prédicat LIKE.

Observons encore que la notion d'atomicité est tout à fait relative et élastique, même si « atomique » veut dire : non décomposable, indivisible. Considérons par exemple la colonne Mbrld, qui est du type INTEGER. Quid si Mbrld vaut 42 ? La valeur 42 est-elle atomique ? On pourrait pinailler et répondre négativement, car 42 est décomposable en facteurs premiers : $42 = 1 \times 2 \times 3 \times 7$. De la même façon, la chaîne de caractères "Developpez.com" peut être considérée comme étant composée de la séquence de caractères : "D,e,v,e,l,o,p,p,e,z,,c,o,m". Pour ne pas s'embarquer dans la controverse on peut préférer utiliser un autre adjectif, en remplaçant « atomique » par « **scalaire** » (voire « **encapsulé** », mais ce terme est connoté Orienté Objet) signifiant que le type (domaine), la colonne, la valeur, la variable, n'ont pas de composants directement visibles, mais accessibles, grâce à des opérateurs *ad hoc*, fournis avec le type, point barre. Par contraste, l'en-tête d'une relation n'est pas scalaire, puisque ses composants (ses attributs) sont visibles par l'utilisateur : la relation et le tuple (n-uplet) sont à ce jour **les seuls objets** qui, du point de vue du Modèle relationnel, font légalement l'objet de types **non scalaires** (d'où un aménagement de la définition de la 1NF, cf. paragraphe 2.7). En tout état de cause, les types ARRAY et MULTISSET de SQL n'en font pas partie, mais pourraient très bien être pris en compte comme cela est précisé dans les remarques concernant Tutorial D (cf. le paragraphe A en annexe).

1.6. Début des années quatre-vingt-dix. Les RVA (attributs dont les valeurs sont des relations)

Quand Codd parle en logicien et écrit en 1970 qu'un calcul du premier ordre est suffisant pour manipuler toutes les relations possibles, dans tous les sens et jusqu'à plus soif (autrement dit en vertu des principes de **complétude** et de **fermeture**), alors que l'année précédente il avait écrit qu'il partait sur la base d'un calcul du deuxième ordre (cf. le paragraphe C en annexe), il s'ensuit que l'*homo relationalis* aimerait connaître les raisons exactes de ce revirement. Malheureusement, le père du Modèle relationnel nous a quittés et nous ne pouvons que conjecturer.

On a vu par ailleurs des gens comme Roth, Korth et Silberschatz se comporter en « hérétiques » et proposer d'étendre le Modèle relationnel grâce notamment à de nouveaux opérateurs (en particulier NEST/UNNEST), permettant d'atomiser récursivement des relations NF²

(Non First Normal Form), relations dont les valeurs peuvent être des relations, à une profondeur quelconque. De leur côté, les gardiens du phare, Date et Darwen, n'ont pas manqué eux aussi d'étudier la possibilité de manipuler des relations contenant d'autres relations, mais tout en respectant la 1NF (aménagée).

Au début des années quatre-vingt-dix, dans un article intitulé « *Relation-Valued Attributes or Will the Real First Normal Form Please Stand Up?* » [Date 1992], D & D se sont livrés à une réflexion approfondie sur la véritable nature de la première forme normale :

« Nous examinons la véritable nature de la première forme normale, et avançons qu'il y a un certain nombre d'avantages à autoriser les attributs des relations à contenir des valeurs qui soient à leur tour (qui encapsulent) des relations. »

Leur constat fut — pour faire court — qu'à condition de laisser tomber la récursivité (cf. [Date 2007a], pages 126-127) et d'en rester aux opérateurs déjà existants, on pouvait utiliser avec profit des **RVA (relation-valued attributes)**, sans avoir à étendre le Modèle relationnel (évolution, oui, mais révolution, non), tout en restant dans le cadre de la première forme normale, dont la définition évolue malgré tout (cf. paragraphe 2.7), comme on l'a déjà brièvement évoqué. Par scrupule, Date a consulté les logiciens, mais personne n'a pu lui montrer que l'utilisation des RVA nécessitait d'en passer par une logique du deuxième ordre (cf. [Date 2007b], page 394). En passant, insistons sur le fait que les RVA ne contiennent que des **valeurs** (relations), jamais de **variables** (relvars), ces choses que le deuxième ordre permet de quantifier.

Pour traiter des RVA, rappelons que, outre les opérateurs de base bien connus : RESTRICT, PROJECT, JOIN etc., D & D ont utilisé deux opérateurs faisant partie depuis une vingtaine d'années du Modèle relationnel (et provenant du langage relationnel ISBL, lequel date des années soixante-dix) : **RENAME** et **EXTEND**. On trouvera dans un article rédigé par Darwen en 2005 (« *HAVING A Blunderful Time (Lien91)* ») l'importance du rôle joué par ces deux opérateurs, que nous allons être amenés à utiliser (concernant leur notation, se reporter au paragraphe B en annexe).

Exemple

Considérons les trois variables relationnelles sans prétention, qui constituent les piliers de la base de données relationnelle « Fournisseurs et Pièces » chère à Chris Date (base de données que nous francisons ici). La variable F est utilisée pour décrire les fournisseurs, la variable P concerne les pièces et la variable FP des liens qui unissent fournisseurs et pièces (« Quels fournisseurs ont livré quelles pièces », « Quelles pièces ont été livrées par quels fournisseurs »).

F				FP		
Four_No	Four_Nom	Statut	Ville	Four_No	Piece_No	Quantite
S1	Salsa	20	Lille	S1	P1	300
S2	Jean	10	Paris	S1	P2	200
S3	Bernard	30	Paris	S1	P3	400
S4	Catherine	20	Lille	S1	P4	200
S5	Alain	30	Arles	S1	P5	100
				S1	P6	100
				S2	P1	300
				S2	P2	400
				S3	P2	200
				S4	P2	200
				S4	P4	300
				S4	P5	400

P				
Piece_No	Piece_Nom	Couleur	Poids	Ville
P1	Ecrou	Rouge	12,0	Lille
P2	Boulon	Vert	17,0	Paris
P3	Vis	Bleu	17,0	Orsay
P4	Vis	Rouge	14,0	Lille
P5	Clou	Bleu	12,0	Paris
P6	Rouage	Rouge	19,0	Lille

Figure 2.6 - Base de données des fournisseurs, des pièces et des livraisons

On trouve en annexe (paragraphe B), l'exemple le plus simple qui soit de l'utilisation de l'opérateur **EXTEND** pour produire à partir d'une relation R une relation R' dont l'en-tête est celui de R, augmenté d'un attribut.

Mais on peut aller beaucoup plus loin et produire une relation dont un attribut prend des valeurs qui sont des relations. Considérons en ce sens l'expression suivante, écrite en Tutorial D, utilisée pour présenter chaque fournisseur avec un inventaire des pièces qu'il fournit, le résultat étant illustré à l'aide de la Figure 2.7 :

```
WITH
  (F RENAME (Four_No AS Four_No_Bis) AS T1),
  (EXTEND T1 ADD (FP WHERE Four_No =
Four_No_Bis) {Piece_No, Quantite} AS Piece_Qte)
AS T2 :
  T2 RENAME (Four_No_Bis AS Four_No) AS F_EXT
```

La construction

```
WITH <liste d'opérations> : <expression relationnelle>
```

permet de fournir sous forme d'une liste d'éléments séparés par des virgules, les opérations que l'on souhaite exécuter en séquence, pour produire au final la relation définie par <expression relationnelle>. La finalité de la liste étant de simplifier l'organisation des opérations qui, à défaut, pourraient devenir compliquées à exprimer.

En l'occurrence, on produit d'abord une relation T1 qui ne diffère de la relation F que par le nom de l'attribut Four_No_Bis. On procède ensuite à l'extension de T1 en récupérant dans FP l'ensemble des tuples pour lesquels la valeur de l'attribut Four_No est égale à celle de l'attribut Four_No_Bis de la relation T1, puis en effectuant une projection sur les attributs Piece_No et Quantite et en nommant Piece_Qte l'attribut (du type relation) obtenu par extension. La relation obtenue est nommée T2. Le résultat final F_EXT est celui de T2, en renommant pour finir Four_No_Bis en Four_No.

La représentation graphique de la relation F_EXT est la suivante :

Four_No	Four_Nom	Statut	Ville	Piece_Qte	
S1	Salsa	20	Lille	Piece_No	Quantite
				P1	300
				P2	200
				P3	400
				P4	200
				P5	100
P6	100				
S2	Jean	10	Paris	Piece_No	Quantite
				P1	300
P2	400				
S3	Bernard	30	Paris	Piece_No	Quantite
P2	200				
S4	Catherine	20	Lille	Piece_No	Quantite
				P2	200
				P4	300
P5	400				
S5	Alain	30	Arles	Piece_No	Quantite

Figure 2.7 - Relation avec attribut à valeur relation (RVA)

Dans la relation F_EXT, les attributs Piece_No et Quantite ont été remplacés par le seul attribut Piece_Qte. Les autres attributs n'ont pas changé. Chaque valeur prise par l'attribut Piece_Qte est une relation (donc en-tête compris), une relation par attribut et par tuple de F_EXT : chaque tuple de F_EXT contient exactement une valeur pour chacun de ses attributs et cette relation est en première forme normale (cf. paragraphe 2.7). Noter que, pour le fournisseur S5, le corps de la relation {Piece_No, Quantite} est l'ensemble vide {}.

En Tutorial D, on pourrait définir un relvar F_EXT :

```
VAR F_EXT BASE RELATION
{
  Four_No      Integer,
  Four_Nom     CHAR,
  Statut       INTEGER,
  Ville        CHAR,
  Piece_Qte    RELATION {Piece_No INTEGER,
Quantite INTEGER}
} KEY {Four_No} ;
```

Intérêt des RVA

Une relation comme F_EXT présente des avantages indéniables :

- Dans la Figure 2.7, le fait que le fournisseur S5 (Alain) n'a pas livré de pièces est représenté par l'ensemble vide au sein de l'attribut Piece_Qte de la relation F_EXT, alors que — en se plaçant dans un contexte SQL — si l'on utilisait une jointure externe (LEFT OUTER JOIN), le résultat ci-dessous serait pollué par le bonhomme NULL et ne pourrait même pas être doté d'une clé, en l'occurrence la paire {Four_No, Piece_No} : F_EXT ne pourrait donc être une relation et le principe de fermeture ne serait pas respecté (horresco referens !) :

Four_No	Four_Nom	Statut	Ville	Piece_No	Quantite
S1	Salsa	20	Lille	P1	300
S1	Salsa	20	Lille	P2	200
S1	Salsa	20	Lille	P3	400
S1	Salsa	20	Lille	P4	200
S1	Salsa	20	Lille	P5	100
S1	Salsa	20	Lille	P6	100
S2	Jean	10	Paris	P1	300
S2	Jean	10	Lille	P2	400
S3	Bernard	30	Paris	P2	200
S4	Catherine	20	Lille	P2	200
S4	Catherine	20	Lille	P4	300
S4	Catherine	20	Lille	P5	400
S5	Alain	30	Arles	NULL	NULL

Figure 2.8 - LEFT OUTER JOIN de F et FP

- En comparant la Figure 2.7 et la Figure 2.8, on se rend compte que dans le premier cas, la redondance est absente, alors qu'elle foisonne dans le second cas.
- La Figure 2.7 correspond à une relation peut-être plus proche de la vision qu'ont certains utilisateurs (ceux pour lesquels, sémantiquement parlant, la relation FP de la Figure 2.6 est une propriété multivaluée de la relation F, plutôt que de la relation P).

Dans leur article mentionné ci-dessus « *Relation-Valued Attributes or Will the Real First Normal Form Please Stand Up?* », D & D fournissent nombre d'observations supplémentaires portant sur l'intérêt des RVA.

Opérateurs GROUP et UNGROUP

Notons en passant que D & D ont prévu deux opérateurs non indispensables (en effet, ce sont des combinaisons d'autres opérateurs), mais permettant de se simplifier la vie pour effectuer des requêtes dans lesquelles interviennent des RVA, à savoir GROUP et UNGROUP. Ainsi, pour obtenir l'équivalent RVA de la relvar FP de la Figure 2.6 :

```
FP GROUP {Piece_No, Quantite} AS Piece_Qte
```

La relation obtenue est représentée dans la Figure 2.9. Inversement, si FPQ désigne cette relation, on peut lui faire subir une opération de dégroupement et retrouver FP :

```
FPQ UNGROUP Piece_Qte
```

Four_No	Piece_Qte	
S1	Piece_No	Quantite
	P1	300
	P2	200
	P3	400
	P4	200
	P5	100
S2	Piece_No	Quantite
	P1	300
S3	Piece_No	Quantite
	P2	200
S4	Piece_No	Quantite
	P2	200
	P4	300
	P5	400

Figure 2.9 - Groupement de FP par Four_No

Inconvénients des RVA

La relation F_EXT (Figure 2.7) est la source d'un phénomène d'**asymétrie** fort gênant, empoisonnant la vie des utilisateurs d'un SGBD hiérarchique comme IMS/DL1, avec lequel on représente *de facto* les données à la manière des RVA.

Par exemple, si l'on considère les requêtes :

1. Quels fournisseurs ont livré la pièce P2 ?
2. Quels sont les pièces livrées par le fournisseur S2 ?

Dans un contexte symétrique, elles sont traduites ainsi (en Tutorial D) :

```
1. (FP WHERE Piece_No = Piece_No ('P2'))
{Four_No}
2. (FP WHERE Four_No = Four_No ('S2'))
{Piece_No}
```

Tandis que dans un contexte asymétrique, elles donnent lieu à celles-ci :

```
1. ((F_EXT UNGROUP (Piece_Qte)) WHERE
Piece_No = Piece_No ('P2')) {Four_No}
2. ((F_EXT WHERE Four_No = Four_No
('S2')) UNGROUP ((Piece_Qte)) {Piece_No}
```

Quant aux contraintes du genre intégrité référentielle, les choses peuvent se compliquer. Si définir une clé étrangère entre les relvars FP et P est chose simple :

```
FOREIGN KEY {Piece_No} REFERENCES P (Piece_No)
```

Entre F_EXT et P, ça l'est moins :

```
(F_EXT UNGROUP (Piece_Qte)) {Piece_No} ⊆ P
{Piece_No}
```

Considérons maintenant les opérations de mise à jour. Exemple :

1. Ajouter dans la base de données le fait que le fournisseur S5 a livré la pièce P5 en quantité 100.
2. Ajouter dans la base de données le fait que le fournisseur S2 a livré la pièce P5 en quantité 200.

Dans un contexte symétrique, elles sont traduites ainsi :

```
1. INSERT FP RELATION {TUPLE {Four_No
Four_No ('S5'),
Piece_No Piece_No ('P5'),
Quantite Quantite (100)}} ;
2. INSERT FP RELATION {TUPLE {Four_No
Four_No ('S2'),
Piece_No Piece_No ('P5'),
Quantite Quantite (200)}} ;
```

Dans un contexte asymétrique, l'affaire se corse :

```
1. INSERT F_EXT RELATION {TUPLE {Four_No
Four_No ('S5'),
Piece_Qte RELATION {TUPLE {Piece_No
Piece_No ('P5'),
Quantite Quantite (100)}}}} ;
2. UPDATE F_EXT WHERE Four_No = Four_No
('S2')
(ININSERT Piece_Qte RELATION {TUPLE {Piece_No
Piece_No ('P5'),
Quantite Quantite (200)}}) ;
```

Etc.

Retrouvez l'article complet de François de Sainte Marie en ligne : [Lien92](#)

IRT : un ray tracer interactif - Partie 2

Deuxième partie de la série de tutoriels dédiée au raytracing (table des matières : [Lien93](#)).
Il est temps d'ajouter la gestion des reflets et des lumières.

1. Objectifs

Les objectifs de ce deuxième tutoriel sont les suivants :

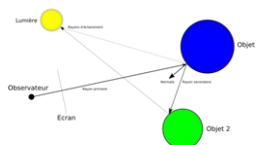
- Gérer les reflets
- Gérer les lumières

Ces deux étapes sont primordiales dans la conception d'un raytracer. Ce sont elles qui permettront d'obtenir un rendu réaliste.

L'encapsulation Python n'a pas beaucoup été modifiée. Les différences proviennent principalement de la gestion de la rotation au tour de l'axe vertical de la scène et de la rotation de la caméra autour de la direction de celle-ci. Ce code n'est pas primordial à la compréhension de cette partie et a donc été passé sous silence.

2. Gestion des reflets

Lorsqu'un rayon lumineux frappe un objet, celui-ci est réfléchi et continue à se propager dans la scène. Ici, on reprend l'approximation de *Whitted*. Il est à l'origine de l'essor du raytracing en approximation la propagation des ondes lumineuses. Lorsque l'une d'elles frappe un objet, 3 types de rayons peuvent être générés : une réflexion, une réfraction et des ombres. Ces dernières font l'objet du prochain point, les réfractions étant pour le moment oubliées.



Réflexion des rayons sur les objets

Chaque objet reflète plus ou moins la lumière d'un autre. Le miroir la reflète totalement, un objet mat ne reflètera rien. Il faut aussi noter que chaque réflexion lumineuse est un nouveau lancer de rayon, donc potentiellement coûteux. En général, on arrête la recherche des réflexions au bout de quelques itérations.

Le rayon lumineux réfléchi se calcule par rapport à la normale à l'objet. En effet, il s'agit simplement de la symétrie du rayon incident par rapport à cette normale.

Tous ces éléments (normale, couleur de l'objet, comment l'objet reflète la lumière) sont stockés dans une structure spécifique :

```
primitives.h
// Carateristique d'une primitive en un
point donné
struct MaterialPoint
{
// Normale de la primitive en un point
Normal3df normal;
// Couleur du point
Color color;
// "Couleur" de la réflexion
Color reflect;
};
```

Ici, le coefficient de réflexion est un vecteur lui-aussi. Le choix a été fait de permettre à un objet d'être un quasi-miroir pour certaines couleurs (valeur de la composante proche de 1) et d'être complètement mat pour d'autres (valeur de la composante proche de 0). Ce choix n'a pas d'incidence sur le code et le nombre de multiplications n'augmente pas, donc l'impact est quasiment nul, tout en ouvrant de nouvelles perspectives (on peut par exemple penser à l'utilisation du raytracer en acoustique où le nombre de composantes de "couleur" est bien plus important et correspondant à des bandes de fréquences sonores et où un obstacle peut absorber ou réfléchir différemment selon la fréquence du rayon).

Une fois cette structure en place, il suffit de modifier la fonction principale du raytracer :

```
raytracer.cpp
void Raytracer::computeColor(const Ray& ray,
Color& color, unsigned int level) const
{
float dist;
long index = scene->getFirstCollision(ray,
dist);
if(index < 0)
return;

Primitive* primitive = scene-
>getPrimitive(index);
MaterialPoint characteristics;
primitive->computeColorNormal(ray, dist,
characteristics);
color = characteristics.color;

if(level < levels)
{
Ray ray_sec(ray.origin() + dist *
ray.direction(), ray.direction() -
(ray.direction() * characteristics.normal) * 2 *
characteristics.normal);
```

```

    ray_sec.direction() *= 1./
(sqrt(norm2(ray_sec.direction())));
    Color color_sec(0.);
    computeColor(ray_sec, color_sec, level+1);

    color += mult(color_sec,
characteristics.reflect);
}
}

```

Si on exécute le code ci-dessous avec la méthode **SimpleScene::getFirstCollision** présentée lors du tutoriel précédent, des erreurs vont apparaître. En effet, lors de la réflexion, on teste si le rayon réfléchi touche d'autres objets. Le problème est qu'il peut commencer par toucher... l'objet d'où il part. Cela est dû aux approximations numériques, donc pour valider la collision entre un rayon et un objet, on ajoute une condition sur la distance de l'objet : celle-ci doit être supérieure à une valeur strictement positive.

```

scene.cpp
long SimpleScene::getFirstCollision(const Ray&
ray, float& dist)
{
    float min_dist =
std::numeric_limits<float>::max();
    long min_primitive = -1;

    for(std::vector<Primitive*>::const_iterator it
= primitives.begin(); it != primitives.end(); +
+it)
    {
        float dist;
        bool test = (*it)->intersect(ray, dist);

        if(test && (0.01f < dist) && (dist <
min_dist))
        {
            min_primitive = it - primitives.begin();
            min_dist = dist;
        }
    }

    if(min_primitive == -1)
        return -1;
    else
    {
        dist = min_dist;
        return min_primitive;
    }
}

```

Maintenant, lorsqu'un rayon touche une primitive, on passe une structure de matériau que la primitive remplira lors du calcul de la normale. La couleur vue sans compter les prochaines réflexions sera directement celle de l'objet.

L'argument **level** indique le nombre de niveaux de réflexion atteint. Si ce nombre est plus petit que la variable constante **levels**, on calcule le rayon réfléchi et on recherche la couleur du rayon qui est reflété sur l'objet. Par la suite, on ajoute à la couleur vue une fraction de cette dernière couleur, selon que l'objet reflète plus ou moins bien.

3. Gestion des lumières

Sans lumière, il n'y a pas d'ombre, les objets restent uniformément "éclairés", fades, ... La lumière dans une scène est émise par plusieurs sources lumineuses. Lorsqu'un rayon de la caméra frappe un objet, le point d'intersection peut être éclairé par une ou plusieurs lumières. Chacune d'elles contribue un peu à la couleur qui sera effectivement vue. Si la source lumineuse est perpendiculaire à l'objet au point d'intersection, la lumière captée sera maximale. Si elle est orthogonale, la lumière sera nulle. On utilise donc le produit scalaire entre la direction de la source lumineuse et la normale pour calculer la quantité de lumière obtenue.

On va donc tester chaque lumière, l'une après l'autre. Si aucun objet n'est entre l'objet et la lumière (ceci étant symbolisé par un rayon partant de l'objet vers la lumière avec une distance maximale), on ajoute la contribution de cette lumière.

```

simple_scene.cpp
const Color SimpleScene::computeColor(const
Point3df& center, const MaterialPoint&
characteristics)
{
    Color t_color(0.);
    for(std::vector<Light*>::const_iterator it =
lights.begin(); it != lights.end(); ++it)
    {
        Vector3df path = (*it)->getCenter() -
center;
        float pathSize = sqrt(norm2(path));
        path.normalize();
        Ray ray(center, path);
        if(testCollision(ray, pathSize))
            continue;

        float cosphi = (path *
characteristics.normal);
        if(cosphi < 0.)
            continue;
        t_color += mult((characteristics.color *
cosphi), (*it)->computeColor(ray, pathSize));
    }

    bool SimpleScene::testCollision(const Ray& ray,
float dist)
    {
        for(std::vector<Primitive*>::const_iterator
it = primitives.begin(); it != primitives.end();
++it)
        {
            float t_dist;
            bool test = (*it)->intersect(ray, t_dist);

            if(test && (0.01f < t_dist) && (t_dist <
dist))
            {
                return true;
            }
        }
        return false;
    }
}

```

Il ne reste plus qu'à donner le code d'une lumière :

light.cpp

```
Color Light::computeColor(const Ray& ray, float
dist)
{
    return color * (1. / (dist * dist));
}

const Vector3df& Light::getCenter() const
{
    return center;
}
```

La quantité de lumière émise par une lumière ponctuelle est constante sur la surface d'une sphère centrée sur cette lumière. La lumière atteignant un point sur cette sphère est donc fonction de l'inverse du rayon au carré.

Le code de la fonction principale du raytracer doit maintenant être modifiée pour que la couleur de l'objet lors du calcul prenne en compte les lumières de la scène :

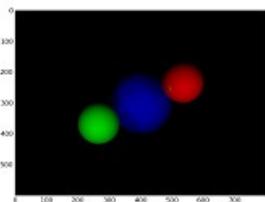
raytracer.cpp

```
void Raytracer::computeColor(const Ray& ray,
Color& color, unsigned int level) const
{
    // ...
    color = scene->computeColor(ray.origin() +
dist * ray.direction(), characteristics);
    // ...
}
```

4. Résultats

4.1. Résultat brut

Voici le résultat affiché par le ray tracer :



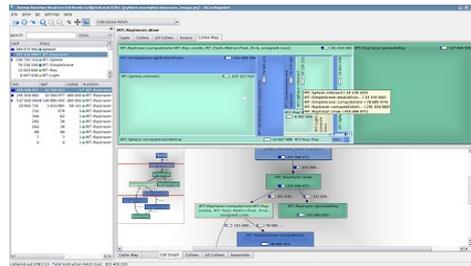
La scène avec reflets et lumières

La figure s'affiche différemment dans l'article précédent. Cette modification d'orientation selon l'axe y a été effectuée pour une compatibilité avec un affichage sur un canevas OpenGL (ce qui sera vu dans un prochain tutoriel). Au fur et à mesure de l'avancée du projet, certains aspects pourront être modifiés.

4.2. Profil du raytracer

Le ray tracer trace cette simple scène en 170 ms sur un Xeon 3.8GHz sous Linux avec GCC 4.1.2.

Voici donc ce profil en mode optimisé calculé par Valgrind ([Lien94](#)), sous Linux :



Profil de la fonction principale draw()

Par rapport au profil de la première partie, un temps plus important est passé dans la méthode de génération des rayons **Raytracer::generateRay**. Effectivement, à cause de la prise en compte des orientations, celle-ci est bien plus complexe. Il s'agit vraisemblablement de la méthode qui bénéficiera le plus d'une optimisation.

On note aussi la récursion des appels à la méthode **Raytracer::computeColor**.

Grâce à un article précédent dédié à la compréhension des profils ([Lien95](#)), de nombreuses autres fonctions ont été optimisées. Par exemple la méthode virtuelle **Sphere::intersect** voit son temps d'exécution diminuer de 40%. A cause de la récursion introduite par les reflets, le temps global d'utilisation a naturellement augmenté, mais sans cette amélioration, le temps global d'exécution serait supérieur à 200ms.

5. Conclusion

La gestion des reflets et des lumières n'est pas terminée ici. En effet, que ce passe-t-il lorsqu'on veut ajouter la gestion de la transparence ? Et comment faire lorsque l'objet contient une fois une texture, l'autre fois une couleur, lorsqu'il contient une carte spécifique pour la normale, ... ?

La solution réside dans une profonde modification du code qui sera à venir. Elle consiste à associer à chaque objet le code qui gèrera aussi le calcul complet de la couleur. Pour l'instant, c'est la classe **Raytracer** qui s'en occupe. A terme, chaque primitive aura son propre calcul, avec gestion des reflets, de la transparence, des lumières, des textures, ... et donc une liste de fonctions usuelles et un champ additionnel contenant ce dont le calcul aura besoin (c'est l'une des seules fois où un pointeur **void*** me sera utile...).

Ces changements étant prévus, je n'ai pas ajouté de paragraphe sur l'optimisation. Ce sera le cas par la suite lorsque les structures d'accélération seront introduites. Mais cela n'est pas pour tout de suite. Avant d'aborder le suréchantillonnage, je présenterai l'encapsulation de la bibliothèque dans une GUI en Python.

Retrouvez l'article de Matthieu Brucher en ligne : [Lien96](#)

1. Tutoriel 11 : Vertex shaders



1.1. Résumé

Ce tutoriel traite des possibilités du vertex shader. Il est conçu pour montrer à l'utilisateur les capacités de manipulation des sommets. A la fin de ce tutoriel, nous aurons un effet de vague imposé au modèle du personnage du tutoriel précédent. Cet effet est créé entièrement par le GPU avec aucune interaction du CPU avec les données.

1.2. Source

(SDK root)\Samples\C++\Direct3D10\Tutorials\Tutorial11

1.3. Shader de sommet

Les tutoriaux précédents ont déjà montrés l'utilisation du vertex shader, ce tutoriel va accentuer ce stage dans le pipeline. Les sommets contiennent différents types d'informations telles que les coordonnées, les normales, les coordonnées de texture, les matériaux, les couleurs, et même des données personnalisées. Par exemple, si vous regardez l'exemple ParticlesGS, vous verrez que chaque sommet passé dans le pipeline contiendra aussi une variable appelée *Timer*, laquelle peut faire varier l'apparence des particules sur le temps.

Le but du vertex shader est de décharger les calculs intensifs du CPU, vers le GPU. Ceci libère le CPU pour accomplir d'autres tâches dans l'application. Par exemple, les jeux déchargent parfois les traitements graphiques vers le GPU, pendant que l'intelligence artificielle et la physique sont effectuées par le CPU.

Ce tutoriel est conçu pour montrer les possibilités du vertex shader, plutôt que d'apprendre des techniques spécifiques. Il est encouragé d'essayer de nouveaux effets et de faire des essais en fonction des résultats. Pour les méthodes spécifiques pour accomplir les effets, se référer aux exemples inclus avec le SDK. Cependant, notez qu'ils sont bien plus complexes et appliquent des concepts bien plus avancés.

1.4. Création de l'effet de vague

L'effet de vague sur le modèle est créé en modulant la position de x de chaque point par une onde sinusoïdale. L'onde sinusoïdale est contrôlée par la position y, de ce fait créant une vague le long du modèle. Cependant, pour faire bouger la vague et donner l'apparence d'animation, l'onde sinusoïdale est également modulée par une variable de temps.

Finalement, il y a une variable pour contrôler le nombre de déplacement par le vertex shader, et ceci est la variable

d'ondulation. Cette variable est passée dans le shader et contrôlée par un curseur graphique.

```
output.Pos.x += sin( output.Pos.y*0.1f + Time )
*Waviness;
```

Après que cette manipulation soit effectuée, le vertex shader devra toujours préparer les sommets pour affichage. Donc, il y a toujours la transformation de la matrice de monde, de la matrice de vue, et la matrice de projection. Cependant, on choisit de faire la transformation de monde avant l'effet de vague, ainsi il sera plus facile de déterminer les effets sur l'écran, car les axes x et y sont apparents.

```
output.Pos = mul( output.Pos, View );
output.Pos = mul( output.Pos, Projection );
```

2. Tutoriel 12 : Pixel shaders



2.1. Résumé

Ce tutoriel se concentre sur le pixel shader et ses possibilités. Il y a plusieurs choses qui peuvent être accomplies avec le pixel shader, et quelques unes des fonctions les plus habituelles sont listées. Le pixel shader appliquera un *environment map* à l'objet.

À la fin de ce tutoriel, le modèle deviendra brillant, et il reflètera son environnement entourant.

2.2. Source

SDK root\Samples\C++\Direct3D10\Tutorials\Tutorial12

2.3. Pixel shader

Les pixel shaders sont utilisés pour manipuler la couleur finale d'un pixel avant qu'il n'atteigne l'*output merger*. Chaque pixel qui est affiché sera passé au travers de ce shader avant la sortie. Une fois que le pixel est passé au travers du pixel shader, les seules opérations qui peuvent être effectuées sont celles effectuées par l'*output merger*, telles que l'*alpha blending*, le test de profondeur, le *stencil testing*, et ainsi de suite.

Le pixel shader a évolué depuis le mapping de texture que l'on trouvait dans les hardwares précédents. Au lieu d'un simple texture de consultation, il est possible de calculer la couleur finale depuis des sources multiples, ainsi que de l'altérer selon les données du sommet. Pour des applications générales, cependant, un pixel shader effectuée de multiples consultations sur différentes textures.

2.4. Environment Map

Sur des surfaces réfléchissantes, il n'est pas possible de juste faire une texture de consultation sur des coordonnées de texture fixes pour déterminer la couleur finale. Cela est dû au fait que quand l'objet ou le spectateur bouge, le reflet change. Donc, nous devons mettre à jour la réflexion chaque fois que quelque chose bouge.

Une méthode efficace pour faire croire à l'observateur qu'il y a une réflexion dynamique sur l'environnement est de générer une texture spéciale qui entoure l'objet. Cette texture s'appelle un *cube map*. Un *cube map* est effectivement de placer un objet au milieu d'un cube, avec chaque face du cube devenant une texture. Un *environment map* est un *cube map* qui a des textures qui correspondent à la vue de l'environnement sur cette face. Si l'environnement est statique, ces *environment maps* peuvent être pré-générés. Si l'environnement est dynamique, le *cube map* doit être mis à jour à la volée. Voir l'exemple CubeMapGS pour une illustration de comment effectuer ceci.

Depuis cet *environment map*, on peut calculer ce qu'une caméra verra comme reflet. D'abord on peut trouver la direction par laquelle la caméra voit l'objet, et de cela, refléter la normale de chaque pixel et effectuer une consultation basé sur ce vecteur reflété.

2.5. Configurer l'Environment Map

La configuration de l'*environment map* n'est pas le but de ce tutoriel. La procédure à suivre est très similaire à celle d'une map texture normale. Voir le Tutoriel 7, **Mapping de texture et Constant Buffers** ([Lien97](#)), pour une explication de comment initialiser correctement une *texture map* et ses vues ressources associées.

```
// Charge l'Environment Map
ID3D10Resource *pResource = NULL;
V_RETURN( DXUTFindDXSDKMediaFileCch( str,
MAX_PATH,
L"Lobby\\LobbyCube.dds" ) );
V_RETURN( D3DX10CreateTextureFromFile( pd3dDevice
, str, NULL, NULL,
&pResource ) );
if(pResource)
{
    g_pEnvMap = (ID3D10Texture2D*)pResource;
    pResource->Release();
    D3D10_TEXTURECUBE_DESC desc;
    g_pEnvMap->GetDesc( &desc );
    D3D10_SHADER_RESOURCE_VIEW_DESC SRVDesc;
    ZeroMemory( &SRVDesc, sizeof(SRVDesc) );
    SRVDesc.Format = desc.Format;
    SRVDesc.ViewDimension =
D3D10_SRV_DIMENSION_TEXTURECUBE;
    SRVDesc.TextureCube.MipLevels =
desc.MipLevels;
    SRVDesc.TextureCube.MostDetailedMip = 0;
    V_RETURN(pd3dDevice-
>CreateShaderResourceView( g_pEnvMap, &SRVDesc,
&g_pEnvMapSRV ));
}
// Défini l'Environment Map
g_pEnvMapVariable->SetResource( g_pEnvMapSRV );
```

2.6. Implémenter le Lookup

Nous allons parcourir l'algorithme simple décrit plus haut et effectuer un lookup propre d'un *environment map*. Les calculs sont effectués dans le vertex shader et ensuite interpolés vers le pixel shader pour le lookup. Il est préférable de le calculer dans le vertex shader et interpoler vers le shader de pixel car il y a moins de calculs requis.

Pour calculer le vecteur refléter pour le lookup, nous avons besoins de deux informations. D'abord, la normale du pixel en question, ensuite, la direction de l'oeil.

Car l'opération est effectuée dans l'espace vue (le view space), nous devons transformer la normale du pixel vers l'espace de vue correcte.

```
float3 viewNorm = mul( input.Norm, (float3x3)View
);
```

Ensuite, nous devons trouver la direction de la caméra. Notez, cependant, que nous sommes déjà en view space, et donc, la direction de la caméra est celle-ci sur l'axe Z (0,0,-1.0), car nous regardons directement vers l'objet.

Maintenant que nous avons nos deux pièces d'informations, on peut calculer la réflexion du vecteur avec la commande **reflect**. La variable *viewR* est utilisée pour stocker la réflexion résultante pour le traitement dans le pixel shader.

```
output.ViewR = reflect( viewNorm, float3(0,0,-
1.0) );
```

Une fois que le vecteur correcte a été calculé dans le vertex shader, on peut traiter l'*environment map* dans le pixel shader. Ceci est effectué en appelant une fonction toute faite pour échantillonner l'*environment map* et retourner la valeur couleur depuis cette texture.

```
// Charge la texture environment map
float4 cReflect =
g_txEnvMap.Sample( samLinearClamp, viewR );
```

Car tout est normalisé (et les coordonnées de textures ont une portée de 0 à 1), les coordonnées x et y correspondront directement aux coordonnées de textures requises pour le lookup.

En bonus, on inclut le code pour faire la texture lookup plate original, et on calcule cela en tant que terme diffus. Pour jouer avec le taux de réflexion, vous pouvez moduler **cReflect** par un facteur de mise à l'échelle par rapport à **cDiffuse** et expérimenter avec les résultats. Vous pouvez avoir un personnage qui reflète beaucoup ou un personnage quelque peu terne.

3. Tutoriel 13 : Geometry Shaders



3.1. Résumé

Ce tutoriel va explorer une partie du pipeline graphique

qui n'a pas encore été abordé dans les tutoriels précédent. Nous allons évoquer quelques fonctionnalités basique des geometry shader.

Le résultat de ce tutoriel sera que le modèle comportera une seconde couche extraite du modèle. Notons que le modèle original sera toujours préservé au centre.

3.2. Source

SDK root\Samples\C++\Direct3D10\Tutorials\Tutorial13

3.3. Geometry Shader

Le bénéfice des geometry shader (GS) est qu'ils permettent de manipuler les mailles sur la base par-primitive. Au lieu de lancer un calcul sur chaque sommet individuellement, il y a l'option pour exploiter sur une base par-primitive. C'est-à-dire, des sommets peuvent être passés dedans comme sommet simple, une ligne segment (deux sommets), ou comme triangle (trois sommets).

En permettant la manipulation à un niveau par-primitive, de nouvelles idées peuvent être approchées, et il y a plus d'accès aux données à tenir compte pour cela. Dans ce tutoriel, vous verrez que nous avons calculé la normale pour le visage. En sachant la position de chacun des trois sommets, nous pouvons trouver la normale du visage.

En plus de permettre l'accès à toutes les primitives, le geometry shader peut créer de nouvelles primitives à la volée. Précédemment dans Direct3D, le pipeline graphique pouvait seulement manipuler le contenu existant, et il pourrait amplifier ou desamplifier des données. Le geometry shader dans Direct3D 10 peut lire dedans une simple primitive (avec les primitives bord-adjacents facultatifs) et émettre zéro, une, ou de multiples primitives basés sur ça.

Il est possible d'émettre un type différent de géométrie que la source d'entrée. Par exemple, il est possible de lire différents sommets, et de générer des triangles multiples basées sur eux. Ceci permet un éventail de nouvelles idées pouvant être exécuté sur le pipeline graphique sans intervention du CPU.

Le geometry shader existe entre le vertex shader et le pixel shader dans le pipeline graphique. Puisque de nouvelles géométries peuvent être créés potentiellement par le geometry shader, nous devons nous assurer qu'ils sont également correctement transformés dans l'espace de projection avant que nous les passions dans le pixel shader. cela peut être fait par le vertex shader avant qu'il entre dans le geometry shader ou il peut être fait dans le geometry shader lui même.

En conclusion, le rendement du Geometry Shader peut être rerouté à un buffer. Vous pouvez lire dans un groupe de triangles, produire de nouvelles géométries, et les stocker dans un nouveau buffer. Cependant, le concept du flux de sortie est au delà de la portée des tutoriels, et il est mieux expliqué dans les exemples trouvés dans le SDK.

Plusieurs des échantillons trouvé dans le SDK Direct3D 10 illustrent les techniques spécifiques qui peuvent être réalisées avec le geometry shader. Si vous souhaitez un

exemple simple pour commencer, vous pouvez essayer ParticlesGS ([Lien98](#)). ParticlesGS simule et affiche un système de particules dynamique (création, explosion et destruction de particules) entièrement sur le GPU.

3.3.1. Formater un Geometry Shader

Contrairement au Vertex shader et au Pixel shader, le geometry shader ne produit pas nécessairement un nombre statique de sorties par entrée. En soi, le format pour déclarer le shader est différent des deux autres.

Le premier paramètre est *maxvertexcount*. Ce paramètre décrit le nombre maximum de sommets qui peuvent être sorti chaque fois que le shader est lancé. Ceci est suivi du nom du geometry shader, qui a été convenablement appelé GS.

Le nom de fonction est suivi des paramètres passés dans la fonction. Le premier contient le mot-clé triangle, qui spécifie que le Geometry Shader opérera sur des triangles en entrée. Le suivant est le format de sommet, et l'identifiant (avec le nombre signifiant la taille du tableau, 3 pour des triangles, 2 pour un segments). Le second est le format et flux de sortie. *TriangleStream* signifie que la sortie sera dans les triangles (bandes de triangle pour être exactes), puis le format est spécifié dans les chevrons. En conclusion, l'identifiant pour le flux est dénotée.

```
[maxvertexcount(12)]  
void GS( triangle GSPS_INPUT input[3], inout  
TriangleStream<GSPS_INPUT> TriStream )
```

Si des sommets commencent à être émis à un *TriangleStream*, il supposera qu'ils tous sont liés ensemble comme une bande. Pour finir une bande, appelez **RestartStrip** dans le flux. Pour créer une liste de triangle, vous devez veiller que vous appelez bien **RestartStrip** après chaque triangle.

3.3.2. Explosion du modèle

Dans ce tutoriel, nous couvrons les fonctions de base des geometry shader. Pour illustrer cela, nous allons créer un effet d'explosion sur notre modèle. Cet effet est créé par extrusion de chaque sommet dans la direction de la normale du triangle considéré.

Notez qu'un effet similaire a été implémenté dans des tutoriels précédent, par lequel nous expulsions chaque sommet par sa normale, commandé par le glisseur *puffiness*. Ce tutoriel démontre l'usage des information d'un triangle complet pour générer la face normale. la différence d'utiliser la face normale est que vous verrez des lacunes entre les triangles éclatés. Parce que les sommets entre différents triangles sont partagés, ils seront donc passés par deux fois dans le geometry shader. De plus, puisque chaque fois qu'une extrusion est effectuée dans la normale du triangle, par opposition au sommet, les deux sommets finaux peuvent finir vers le haut dans différentes positions.

3.3.3. Calculer la face normale

Pour calculer la normale pour n'importe quel plan, nous avons besoin d'abord de deux vecteurs qui résident sur le plan. Puisque nous nous sommes donnés un triangle, nous

pouvons soustraire deux sommets quelconques du triangle pour obtenir les vecteurs relatifs. Une fois que nous avons les vecteurs, nous prenons le produit en croix pour obtenir la normale. Nous devons également normaliser la normale, puisque nous reverrons l'échelle plus tard.

```
//  
// Calculer la face normale  
//  
float3 faceEdgeA = input[1].Pos -  
input[0].Pos;  
float3 faceEdgeB = input[2].Pos -  
input[0].Pos;  
float3 faceNormal =  
normalize( cross( faceEdgeA, faceEdgeB ) );
```

Une fois que nous avons la normale de la face, nous pouvons expulser chaque point du triangle dans cette direction. Pour faire ainsi, nous employons une boucle, qui fera trois tours et opérera sur chaque sommet. La position du sommet est expulsée par la normale, multipliée par un facteur. Puis, puisque le Vertex shader n'a pas transformé les sommets à l'espace approprié de projection, nous devons également faire cela dans le Geometry shader. En conclusion, une fois que nous empaquetons le reste des données, nous pouvons apposer ce nouveau sommet à notre *TriangleStream*.

```
for( int v=0; v<3; v++ )  
{  
    output.Pos = input[v].Pos +  
float4( faceNormal*Explode, 0 );  
    output.Pos = mul( output.Pos, View );  
    output.Pos = mul( output.Pos,  
Projection );  
  
    output.Norm = input[v].Norm;  
  
    output.Tex = input[v].Tex;  
  
    TriStream.Append( output );  
}
```

Une fois que les trois sommets ont été émis, nous pouvons couper la bande et recommencer. Dans ce tutoriel, nous voulons expulser chaque triangle séparément, ainsi nous finissons avec une liste de triangle.

```
TriStream.RestartStrip();
```

Ce nouveau flux de triangle est alors envoyé au Pixel shader, qui opérera sur ces données et les dessinera à la cible d'affichage.

Retrouvez l'article de la Msdn traduit et relu par Romain Perruchon, Cyril Doillon et Arnaud Feltz en ligne : [Lien99](#)

Les dernières news

E3 2009

Comme vous le savez sûrement, l'E3, le plus célèbre des salons de jeux vidéo au monde a ouvert ses portes début juin à Los Angeles. Microsoft a décidé d'organiser sa conférence de presse au début de l'exposition. Ce fût l'occasion de découvrir les prochaines nouveautés en termes de jeux et de matériels pour la console phare de la marque : la Xbox360.

De nombreuses stars étaient présentes pour présenter les diverses nouveautés comme Steven Spielberg ou les Beatles. Le Xbox live a également eu sa part du gâteau par ces nouvelles fonctionnalités.

Enfin, le principal événement de la conférence fût la présentation du "**Project Natal**" qui a pour objectif de remplacer les manettes par le corps humain.

Le lendemain, Nintendo a décidé d'organiser sa conférence de presse et ce fût l'occasion de découvrir les nouveautés de Big N pour l'année à venir... On notera tout d'abord les annonces de deux nouveaux Mario : *New Super Mario Bros :Wii* et *Super Mario Galaxy 2*.

Coté matériel, Nintendo reste sur des valeurs sûres en prolongeant l'aspect fitness de la console Wii. Un nouveau périphérique se branchant au bout du doigt a fait son apparition : le **Wii Vitality Sensor** qui pourra évaluer vos pulsations cardiaques. Ce composant sera certainement utile pour le prochain programme de coaching Wii Fit Plus.

On remarque immédiatement que Nintendo n'a pas pris de risque sur ses annonces en jouant la sécurité par des valeurs sûres.

Une heure après la fin de la séance de Nintendo, Sony a débuté sa conférence de presse concernant ses nouveautés et orientations pour les mois à venir.

La conférence a commencé, contrairement à ces concurrents, par une présentation des chiffres de vente de ses consoles. Très vite suivi par une présentation très rapide des jeux « Blockbuster » à sortir sur les consoles Playstation.

Une section concernant intégralement la console portable PSP de Sony. Après une présentation succincte de jeux, très rapidement a été officialisé le nouveau modèle de PSP : la **PSP Go** qui se présente plus compact avec un volet coulissant mais sans port UMD. C'est donc sous format "boîte" UMD et téléchargement sur le Playstation Network que sortiront les prochains jeux sur la console de Sony.

Pour suivre la mouvance de ces concurrents, il a été présenté le prototype d'un nouveau contrôleur à détection de mouvement. A mi chemin en la Wiimote et l'Eye Toy, ce périphérique semble d'une précision redoutable sur les démonstrations simple présentées. Annoncé pour printemps 2010, nous verrons si Sony arrive à suivre ses concurrents sur la course des détections de mouvements.

Commentez les news sur l'E3 2009 en ligne : [Lien100](#) – [Lien101](#) - [Lien102](#)

Liens

- Lien1 : <http://www.developpez.net/forums/d730365/club-professionnels-informatique/evolutions-club/nouveau-portail-dynamique-professionnel-communautaire-vos-avis/>
- Lien2 : <http://baptiste-wicht.developpez.com/outils/jetbrains/intellij/idea/6/>
- Lien3 : <http://dejardin.developpez.com/outils/jetbrain/intellij/idea/6/>
- Lien4 : <http://hikage.developpez.com/java/review/jetbrains/intellij/idea/7/>
- Lien5 : <http://account.jetbrains.com/registration.jsp>
- Lien6 : <http://hikage.developpez.com/java/review/jetbrains/intellij/idea/8/>
- Lien7 : <http://www.developpez.net/forums/u37042/hikage/>
- Lien8 : <http://spring.developpez.com/>
- Lien9 : <http://spring.developpez.com/cours/>
- Lien10 : <http://java.developpez.com/faq/spring/>
- Lien11 : <http://java.developpez.com/livres/?page=Spring>
- Lien12 : <http://blog.developpez.com/recap/java/c1820/recapitulatif-java/frameworks/spring/>
- Lien13 : <http://www.developpez.net/forums/d745314/java/general-java/spring/nouvel-espace-dedie-spring-developpez-com/>
- Lien14 : <http://www.manning.com/Struts2inAction>
- Lien15 : <http://java.developpez.com/livres/>
- Lien16 : <http://eric-pommereau.developpez.com/tutoriels/outil-web/firebug/>
- Lien17 : <http://cssglobe.com/post/1960/css-dos-and-donts-part-2-markup>
- Lien18 : <http://cssglobe.developpez.com>
- Lien19 : <http://cssglobe.developpez.com/tutoriels/html/comment-utiliser-titres/>
- Lien20 : <http://cssglobe.developpez.com/tutoriels/css/balises/>
- Lien21 : <http://servebox.developpez.com/tutoriels/as3/actionscript-foundry/gestionnaire-contacts-partie1/concept-de-base/>
- Lien22 : <http://servebox.developpez.com/tutoriels/as3/actionscript-foundry/gestionnaire-contacts-partie1/sources/contact-application-source-partie2-debut.zip>
- Lien23 : <http://servebox.developpez.com/tutoriels/as3/actionscript-foundry/gestionnaire-contacts-partie1/concept-de-base/#L3-3>
- Lien24 : <http://servebox.developpez.com/tutoriels/as3/actionscript-foundry/gestionnaire-contacts-partie2/services-et-searchTree/>
- Lien25 : <http://thisblog.usejquery.com/2009/03/18/create-a-unique-gallery-by-using-z-index-and-jquery/>
- Lien26 : <http://dmouronval.developpez.com/tutoriels/javascript/galerie-images-avec-z-index-et-jquery/fichiers>
- Lien27 : http://dmouronval.developpez.com/tutoriels/javascript/galerie-images-avec-z-index-et-jquery/fichiers/z-index_gallery.zip
- Lien28 : <http://960.gs/>
- Lien29 : <http://www.imagemagick.org/script/index.php>
- Lien30 : https://developer.mozilla.org/en/Canvas_tutorial
- Lien31 : <http://dmouronval.developpez.com/tutoriels/javascript/galerie-images-avec-z-index-et-jquery/>
- Lien32 : <http://thisblog.usejquery.com/2009/03/25/enhancing-the-z-index-gallery-with-a-preloader/>
- Lien33 : <http://dmouronval.developpez.com/tutoriels/javascript/jquery/ameliorer-galerie-z-index-avec-preloader/fichiers>
- Lien34 : http://dmouronval.developpez.com/tutoriels/javascript/jquery/ameliorer-galerie-z-index-avec-preloader/fichiers/z-index_gallery_enhanced.zip
- Lien35 : <http://www.ajaxload.info/>
- Lien36 : <http://dmouronval.developpez.com/tutoriels/javascript/jquery/ameliorer-galerie-z-index-avec-preloader/>
- Lien37 : <http://guides.rubyonrails.org/>
- Lien38 : <http://www.ruby-lang.org/>
- Lien39 : <http://rubygems.rubyforge.org/>
- Lien40 : <http://www.sqlite.org/>
- Lien41 : <http://www.mysql.com/>
- Lien42 : <http://www.postgresql.org/>
- Lien43 : <http://www.infoq.com/articles/rest-introduction>
- Lien44 : <http://docs.rubygems.org/read/chapter/1#page22>
- Lien45 : http://en.wikipedia.org/wiki/Create_read_update_and_delete
- Lien46 : <http://api.rubyonrails.org/>
- Lien47 : <ftp://ftp-developpez.com/v-dubois/rails-chapitre1/mycontacts-chapitre1.zip>
- Lien49 : <http://jquery.com/>
- Lien50 : <http://abeautifulsite.developpez.com/tutoriels/javascript/frameworks/jquery/arborescence-fichiers/fichiers/exemple.htm>
- Lien51 : <http://abeautifulsite.developpez.com/tutoriels/javascript/frameworks/jquery/arborescence-fichiers/fichiers/jqueryFileTree.zip>
- Lien52 : <http://gsgd.co.uk/sandbox/jquery/easing/>
- Lien53 : <http://laviska.com/>
- Lien54 : <http://www.chazzuka.com/>
- Lien55 : <http://carlosgallupa.com/>
- Lien56 : <http://www.marcsabourdin.com/>
- Lien57 : <http://devblog.jasonhuck.com/>
- Lien58 : <http://datahack.se/>
- Lien59 : <http://abeautifulsite.developpez.com/tutoriels/javascript/frameworks/jquery/arborescence-fichiers/>
- Lien60 : <http://www.developpez.net/forums/d749214/c-cpp/cpp/threads-processus/article-intel-software-conference-2009-intel-parallel-studio/>
- Lien61 : <http://www.developpez.net/forums/d749217/c-cpp/cpp/threads-processus/interview-james-reinders-intel-parallelisme/>
- Lien62 : <http://www.developpez.net/forums/d748778/c-cpp/cpp/threads-processus/cpp-adapte-multithreading-programation-parallele/>
- Lien63 : <http://loic-joly.developpez.com/articles/intel-software-conference-2009/>
- Lien64 : <http://www.amazon.fr/exec/obidos/ASIN/0974364959/wwwdeveloppez-21>
- Lien65 : <http://cpp.developpez.com/livres/?page=livresBibliotheques#L9780596514808>
- Lien66 : <http://software.intel.com/en-us/data-parallel/>
- Lien67 : <http://loic-joly.developpez.com/articles/interview-james-reinders/>
- Lien68 : <http://adserver.adtech.de/?adlink%7C3.0%7C224%7C1429628%7C1%7C16%7CAdId%3D2383476%3BBnId%3D1%3Blink%3Dhttp%3A%2F%2Fwww.sosdevelopers.com%2Fparallel-studio.htm>
- Lien69 : http://matthieu-brucher.developpez.com/tutoriels/cpp/profil-valgrind-visual-studio/?page=visual_studio
- Lien70 : <http://matthieu-brucher.developpez.com/tutoriels/cpp/totalview/>

Lien71 : <http://matthieu-brucher.developpez.com/tutoriels/cpp/intel-parallel-studio/>
Lien72 : <http://fablain.developpez.com/tutoriel/initiation-sharepoint/>
Lien73 : <http://www-ipcms.u-strasbg.fr/spip.php?rubrique308.html>
Lien74 : <http://www.developpez.net/forums/d753930/club-professionnels-informatique/actualites/disques-durs-lus-ecrits-100-000-plus-grace-lasers/>
Lien75 : <http://mac.developpez.com/livres/>
Lien76 : http://images.apple.com/euro/macosex/technology/docs/GrandCentral_TB_brief_20090608.pdf
Lien77 : http://images.apple.com/euro/macosex/technology/docs/OpenCL_TB_brief_20090608.pdf
Lien78 : <http://gizmodo.com/5287179/macbook-pro-2009-review>
Lien79 : <http://anandtech.com/mac/showdoc.aspx?i=3580&p=4>
Lien80 : <http://www.laptopmag.com/review/laptops/apple-macbook-pro-15-inch-2009.aspx?page=3>
Lien81 : <http://www.pcmag.com/article2/0,2817,2348660,00.asp>
Lien82 : <http://mac.developpez.com/evenements/wwdc/2009/>
Lien83 : <http://openssh.com/>
Lien84 : <http://www.cygwin.com/>
Lien85 : http://man.developpez.com/man5/sshd_config.5.php
Lien86 : <http://mathieu-androz.developpez.com/articles/linux/ssh/>
Lien87 : http://www.openbsd.org/cgi-bin/man.cgi?query=sshd_config&sektion=5
Lien88 : http://fr.wikipedia.org/wiki/Chiffrement_asymétrique
Lien89 : <http://mah.everybody.org/docs/ssh>
Lien90 : <http://come-david.developpez.com/tutoriels/remoteX/#LIII-b-2>
Lien91 : <http://www.dcs.warwick.ac.uk/~hugh/TTM/HAVING-A-Blunderful-Time.html>
Lien92 : <http://fsmrel.developpez.com/basesrelationnelles/normalisation/>
Lien93 : <http://matthieu-brucher.developpez.com/tutoriels/3D/raytracer/>
Lien94 : <http://valgrind.org/>
Lien95 : <http://matthieu-brucher.developpez.com/tutoriels/cpp/profil-valgrind-visual-studio/>
Lien96 : <http://matthieu-brucher.developpez.com/tutoriels/3D/raytracer/02-reflets/>
Lien97 : http://msdn.developpez.com/direct3d/10/tutoriels/base/?page=page_1_7
Lien98 : [http://msdn.microsoft.com/fr-fr/library/bb205329\(en-us,V.S.85\).aspx](http://msdn.microsoft.com/fr-fr/library/bb205329(en-us,V.S.85).aspx)
Lien99 : <http://msdn.developpez.com/direct3d/10/tutoriels/shader/>
Lien100 : <http://www.developpez.net/forums/d754256/applications/developpement-2d-3d-jeux/e3-2009-microsoft-ouvre-bal/>
Lien101 : <http://www.developpez.net/forums/d755078/applications/developpement-2d-3d-jeux/e3-2009-conference-nintendo-risques/>
Lien102 : <http://www.developpez.net/forums/d755088/applications/developpement-2d-3d-jeux/e3-2009-conference-sony-quelques-surprises/>
Lien103 : <http://www.developpez.net/forums/d769827/java/edi-outils-java/autres-edi/jetbrains-milestone-1-ledi-intellij-idea-9-disponible/>