



Developpez

Le Mag

Édition de Février - Mars 2011.

Numéro 32.

Magazine en ligne gratuit.

Diffusion de copies conformes à l'original autorisée.

Réalisation : Alexandre Pottiez

Rédaction : la rédaction de Developpez

Contact : magazine@redaction-developpez.com

Sommaire

Java	Page 2
Android	Page 6
Spring	Page 9
Zend Framework	Page 12
(X)HTML/CSS	Page 19
Développement Web	Page 22
Webmarketing	Page 26
Perl	Page 28
Qt	Page 35
Windows Phone	Page 45
Mac	Page 49
iOS	Page 51
Mobiles	Page 54
Access	Page 60
XML	Page 66
Liens	Page 74

Article Zend Framework



Installer le framework Zend et créer un projet ZF en 15 minutes max.

Vous verrez dans cet article comment installer le framework Zend et créer un projet ZF, le tout en une quinzaine de minutes.

par **Michael Vaudin**
Page 12



Article Webmarketing

Ajax et le référencement

Cet article constitue une introduction au développement d'applications pour Windows Phone 7. Il est le premier d'une série traitant de divers aspects de programmation sous ce nouvel environnement.

par **Marc Chappuis**
Page 26

Éditorial

Il y a un an, la rubrique Mobiles apparaissait dans ce magazine et ne faisait qu'une page.

Je vous promettais d'étoffer cet aspect du développement.

Chose promise, chose due, vous trouverez ce mois-ci pas moins de seize pages réparties sur quatre rubriques.

Profitez-en bien !

La rédaction



JBoss Application Server 6 disponible

Le serveur d'application Java de Red Hat offre le support complet de Java EE 6

La nouvelle version de JBoss, le serveur d'application Java est disponible.

Il s'agit de l'un des premiers serveurs à offrir un support complet et prêt pour la production de Java Entreprise Edition 6 (JEE 6), la spécification du langage Java qui peine encore à se faire une place dans les entreprises.

JBoss est un projet open-source gratuit, racheté et mené depuis 2006 par Red Hat, qui offre aussi un support payant dans le cadre du package JBoss Enterprise Middleware et de JBoss Enterprise Application Platform.

Pour mémoire, GlassFish, le serveur d'application Java d'Oracle racheté avec Sun, offre aussi un support complet des spécifications de Java EE 6, mais il est présenté par Oracle comme un serveur léger. L'alternative réelle à la technologie de Red Hat serait donc plutôt WebLogic, mais ce dernier n'offre pas encore le support de Java EE 6.

Quant à IBM, la nouvelle version de son serveur d'application WebSphere devrait offrir le support de la dernière version de Java en date (en attendant [Java 7](#)). Mais sa sortie n'est programmée que pour plus tard dans l'année.

Enfin, Apache Tomcat (le serveur d'application Java le plus utilisé) est en phase bêta de sa version 7. Une version qui n'offrira de toute façon qu'un support partiel de Java EE 6.

La version finale de JBoss 6 est disponible en téléchargement sur cette page : [Lien 01](#)

Commentez cette news d'Idelways en ligne : [Lien 02](#)

Oracle va-t-il cantonner le futur de Java aux besoins des grandes entreprises ?

C'est ce que prévoit un rapport de Forrester Research

D'après un rapport de l'institut Forrester Research, Oracle serait en train de restreindre le développement du langage Java pour qu'il réponde aux besoins des entreprises, et ce au détriment du reste de l'écosystème ; car, d'après l'étude, « *c'est là où se trouve l'argent (sic)* ».

Selon Jeffrey Hammond et John Rymer, coauteurs du rapport, Java risque à l'avenir de perdre son titre de langage généraliste le plus populaire au monde ([Lien 03](#)) et de devenir un langage côté serveur spécialisé, servant essentiellement les besoins des clients d'Oracle et de IBM.

Un tableau plutôt sombre que dresse ici Forrester, qui s'appuie sur le fait que Oracle n'a jamais caché l'intérêt lucratif qu'il porte à ce langage. Larry Ellison, son PDG, a qualifié à maintes reprises Java d'élément le plus précieux du rachat de Sun.

Bien que la plupart des spécifications du langage soient sous licence open-source, l'étude rappelle le contrôle étroit que maintient Oracle sur Java. Le numéro 3 mondial du secteur possède aujourd'hui la marque Java et le kit de validation pour la compatibilité des implémentations alternatives. Sans oublier le contrôle plus ou moins ouvert du Java Community Process (JCP).

Jeffrey Hammond et John Rymer citent d'ailleurs le cas de la Fondation Apache, qui semble avoir quitté définitivement ce JCP ([Lien 04](#)). Un départ qui, selon le cabinet d'étude, aurait « *mis à mal la crédibilité d'Oracle en tant que partenaire des "Java alpha geeks" à l'origine de tant d'innovations indépendantes* ».

Autre argument en faveur de l'hypothèse de Forrester Research, Oracle ne s'attaquerait pas à l'une des plus importantes faiblesses de Java : sa complexité.

Une complexité qui pourrait sur le moyen terme inciter les développeurs à envisager des alternatives pour le développement d'applications internes ou destinées au Cloud Computing, comme la plateforme .NET de Microsoft.

Forrester prédit par ailleurs qu'OpenJDK restera le seul et unique moteur d'exécution Java disponible sous licence open-source.

Autre prédiction, Oracle et ses partenaires vont tenter à l'avenir de créer un regain d'intérêt pour Java EE, de plus en plus abandonné au profit du Framework Spring (parmi d'autres frameworks qui servent à affranchir les développeurs de la complexité du langage).

Ce rapport s'appuie sur une enquête menée auprès de 12 organisations directement impliquées dans Java, comme Oracle, IBM, Red Hat mais aussi Microsoft et la fondation Apache. Il prend également en compte les réactions des utilisateurs du blog de l'institut Forrester Research.

Commentez cette news d'Idelways en ligne : [Lien 05](#)

Les exploits sur Java ont dépassé ceux ciblant les produits Adobe en 2010, d'après le Cisco Annual Security Report

Cisco vient de publier un nouveau rapport faisant un bilan des principales menaces numériques, à partir de statistiques recueillies tout au long de l'année 2010.

Première constatation : l'année passée a été la première à voir le volume de pourriels décliner. Cette chute du nombre de spams envoyés varie d'une région à l'autre, elle est par exemple de -1.6 % aux USA (par rapport à 2009) et de -47.5 % au Brésil, ou de -87 % en Turquie.

Malgré tout, le pays le plus touché par ces e-mails indésirables reste les Etats-Unis (avec 11.1 billions d'unités échangées en 2010).

Ce recul pourrait s'expliquer par l'arrestation au cours des douze derniers mois de huit gros expéditeurs de spams.

Mais si le nombre de courriels envoyés s'est réduit, il n'en va pas de même pour les risques, et Cisco préconise de ne pas oublier d'utiliser des filtres pour autant.

Second grand point de l'étude, les vulnérabilités les plus exploitées en 2010 n'étaient pas celles du format PDF d'Adobe (comme beaucoup le pensent) : "Les failles Java exploitées ont dépassé celles de Reader et Acrobat. Java a été 3.5 fois plus exploité que les PDF malicieux".

Plus concrètement, les attaques visant Java n'étaient que de 1.5 % en janvier 2010, pour arriver à 7 % douze mois plus tard (contre 2 % pour Adobe, qui était en revanche à 6 % en janvier).

Des menaces renforcées par les soucis que beaucoup d'utilisateurs de Java ont rencontrés lors de leurs tentatives de mise à jour (leur logiciel leur indiquait qu'il était updaté, alors qu'il ne l'était pas ; aussi lorsque le logiciel mis à jour laisse tout de même en place sur le système une ancienne version).

De plus, tout le monde avait les yeux tournés vers Adobe, ce qui a laissé le champ libre aux pirates. Cela devrait changer en 2011 avec une prise de conscience plus large des dangers menaçant Java.

Commentez cette news d'Erna Katleen en ligne : [Lien 06](#)

Les derniers tutoriels et articles

Afficher des animations Flash dans les composants SWING

Adobe Flash est un logiciel multimédia incontournable qui permet de créer des animations graphiques et des objets interactifs.

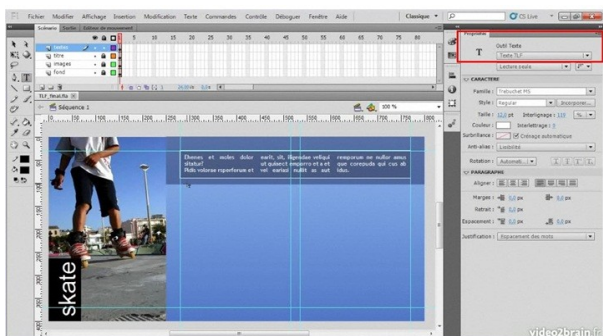
Cet article est destiné aux développeurs Java qui souhaitent ajouter des animations Flash dans leurs composants SWING dans un environnement Windows.

1. Introduction

Pour ceux qui connaissent le logiciel Flash, il est certes aussi facile de créer des animations vectorielles en Flash qu'en Java.

Ces animations peuvent varier d'un simple mouvement rectiligne d'un objet graphique, à la création d'une séquence de dessins animés.

La création de ces animations est rendue facile et rapide grâce à la richesse et la convivialité de l'IDE Flash qui est actuellement en version CS5.



Les animations Flash portent l'extension **.SWF**. Ce format de publication est connu pour sa grande capacité multimédia vu qu'on peut y incorporer des dessins vectoriels, des images, des sons waves ou mp3, comme on

peut aussi ajouter de la vidéo.

Pensez à intégrer ces animations à vos conteneurs SWING sera sûrement une idée originale.

La firme Apple fournit le plug-in QuickTime pour permettre de lire différents formats multimédia notamment les fichiers **.SWF**. Les fonctionnalités sont réduites et les versions récentes de Flash player ne sont pas supportées par ce plug-in, ce qui nous amène à délaisser cette technologie.

Une autre technologie qui peut être utilisée pour afficher les fichiers **.SWF** en Java est l'API **JFlashPlayer** qui permet de plus le contrôle des animations Flash avec des instructions Java.

2. Présentation de l'API JFlashPlayer

Cette API est actuellement dans sa version 2.0. Elle fait partie du paquetage Jpackages ([Lien 07](#)) qui contient une collection de bibliothèques qui peuvent s'interfacer avec les API Java standards.

Cette API fournit plusieurs fonctionnalités pour manipuler les fichiers SWF :

- lire des animations Flash avec un contrôle pour arrêter, rembobiner, lire en boucle, etc ;
- appeler des méthodes Java à partir du code ActionScript (langage script propriétaire fourni avec l'IDE Flash) ;
- appeler des fonctions créées en ActionScript dans

- les classes Java ;
- installer le plug-in Flash player dynamiquement s'il n'existe pas sur la machine au préalable ;
- etc.

Cette API peut être utilisée seulement sous un environnement Windows notamment Windows 98, Me, NT4, 2000, XP et Vista. Les auteurs de l'API ne prévoient pas créer des versions pour les systèmes Linux et Mac OS. On ne peut pas utiliser cette API dans les applets pour plusieurs raisons, y compris le niveau de sécurité fourni par les applets.

3. Etude de cas : intégrer une animation Flash dans une fenêtre Java

3.1. Configurations nécessaires

Pour fonctionner, **JFlashPlayer** exige l'inclusion de bibliothèques natives d'extension **.dll** qui doivent être spécifiées dans la variable d'environnement **PATH**.

Il faut donc installer le **Flash player** sur la machine pour que l'API puisse fonctionner. S'il est déjà installé, ce player peut être détecté, sinon on peut l'installer dynamiquement avec les méthodes fournies par l'API.

Les versions JRE supportées par l'API sont : 1.4, 1.5 et 1.6.

- Sur le site de l'API ([Lien 08](#)), vous trouvez un zip qu'il faut télécharger et décompresser.

Ce paquetage contient une archive **jflashplayer.jar** qui doit être spécifiée dans la **CLASSPATH** de votre IDE Java pour pouvoir travailler avec l'API.

3.2. Implémentation

Dans cette partie, je vais présenter le code d'une application qui permet de lire et contrôler des animations Flash en Java.

Cette application est composée d'une seule classe "**Lecteur**" qui crée une instance de la classe **JFrame** pour afficher une fenêtre dans laquelle on va intégrer une animation Flash.



Au début, il faut créer le menu "**Fichier**" composé de six commandes permettant la lecture et le contrôle du contenu SWF.

```
private void créerMenu()
{
    getContentPane().setBackground(Color.WHITE);
```

```
    setJMenuBar(menuBar);

    menuBar.add(fichierMenu);
    fichierMenu.setText("Fichier");

    fichierMenu.add(menuOuvrir);
    menuOuvrir.addActionListener(this);
    menuOuvrir.setText("Ouvrir");

    fichierMenu.add(menuLire);
    menuLire.setText("Lire");
    menuLire.addActionListener(this);

    fichierMenu.add(menuArrêter);
    menuArrêter.setText("Arrêter");
    menuArrêter.addActionListener(this);

    fichierMenu.add(menuArrière);
    menuArrière.setText("Arrière");
    menuArrière.addActionListener(this);

    fichierMenu.add(menuAvant);
    menuAvant.setText("Avant");
    menuAvant.addActionListener(this);

    fichierMenu.add(menuRebobiner);
    menuRebobiner.setText("Rebobiner");
    menuRebobiner.addActionListener(this);
}
```

Pour gérer les appuis sur ces menus de commandes, on doit redéfinir la méthode **actionPerformed()** de l'interface **ActionListener** :

```
public void actionPerformed(ActionEvent menuSelectionné)
{
    try {
        if (!FlashPanel.hasFlashVersion("10"))

            // si la version 10 du flash player
            // n'existe pas sur la machine
            // alors lancer son installation
            // dynamiquement

            FlashPanel.installFlash("10");

    } catch (JFlashLibraryLoadFailedException err) {

        err.printStackTrace();
    }

    if (menuSelectionné.getSource() == menuOuvrir) {
        JFileChooser fileOuvrirSWF = new
        JFileChooser();
        if (fileOuvrirSWF.showOpenDialog(this) ==
        JFileChooser.APPROVE_OPTION) {

            FlashPanel.setRequiredFlashVersion("10");

            try {

                // le deuxième paramètre est à false pour
                // ne pas démarrer l'animation à son
                // chargement

                lecteurFlash = new FlashPanel(new
```

```

File(fileOuvrirSWF
    .getSelectedFile().getAbsolutePath(), false);
    getContentPane().add(lecteurFlash);
    validate();

    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (JFlashLibraryLoadFailedException e) {
        e.printStackTrace();
    } catch (JFlashInvalidFlashException e) {
        e.printStackTrace();
    }
    }
} else if (menuSelectionné.getSource() ==
menuLire) {

    // lancer la lecture de l'animation flash
    lecteurFlash.play();

    } else if (menuSelectionné.getSource() ==
menuArreter) {

    // arrêter la lecture de l'animation
    lecteurFlash.stop();

    } else if (menuSelectionné.getSource() ==
menuArrière) {

    // reculer la tête de lecture
    lecteurFlash.back();

    } else if (menuSelectionné.getSource() ==
menuAvant) {

    // avancer la tête de lecture instantanément
    lecteurFlash.forward();

    } else if (menuSelectionné.getSource() ==
menuRebombiner) {

    //commencer une nouvelle lecture de l'animation
    lecteurFlash.rewind();

    }
}

```

Il faut spécifier la version du Flash player installée sur la machine avec la méthode **setRequiredFlashVersion()** :

```
FlashPanel.setRequiredFlashVersion("version_flash_palyer");
```

Avec le code fourni en haut, on utilise la version 10. Il faut vérifier la version du player installé sur la machine avec la méthode **hasFlashVersion()**.

```
FlashPanel.hasFlashVersion("version_flash_palyer");
```

Si cette la méthode retourne la valeur false, il faut donc installer cette version.

```
FlashPanel.installFlash("num_version");
```

Il vaudrait mieux travailler avec une version installée au préalable.

On peut donc afficher la liste des Flash player installés :

```

for(int i=0;i<13;i++)
{
    System.out.println("version : "+i+" disponible
? : "+FlashPanel.hasFlashVersion(""+i));
}

```

4. Liens utiles

- Installer le paquetage JFlashPlayer : [Lien 09](#)
- Documentation de l'API : [Lien 10](#)
- Télécharger Adobe Flash Player : [Lien 11](#)
- Adobe Flash : [Lien 12](#)
- Sources de l'application : [Lien 13](#)

5. Conclusion

L'API JFlashPlayer est certainement intéressante. Elle permet en fait de profiter de la puissance de Flash dans la création des animations pour les intégrer dans les interfaces graphiques créées en Java.

Si vous voulez encore publier des fichiers SWF en utilisant des instructions Java, vous pouvez consulter mon article traitant de l'API Transform SWF ([Lien 14](#)).

Retrouvez l'article de Mohamed Slim en ligne : [Lien 15](#)



Introduction aux styles et aux thèmes

Cet article vous présente une particularité assez plaisante d'Android vis-à-vis des autres systèmes d'exploitation mobiles : la possibilité de créer des interfaces graphiques à l'aide de simples fichiers XML.

1. Introduction

De toutes les particularités d'Android vis-à-vis des autres systèmes d'exploitation mobiles, une me plaît tout particulièrement : la possibilité de créer des interfaces graphiques à l'aide de simples fichiers XML. Cette particularité favorise grandement la clarté du code source des applications. En effet, séparer la description de l'interface graphique (layout XML) de la logique applicative (code Java) est en gage d'extensibilité.

Etant très attaché aux technologies du Web (XHTML, JavaScript, CSS, *etc.*), j'ai souvent tendance à comparer la création d'applications au développement d'un site Internet.

Ainsi :

- les layouts s'apparentent au contenu XHTML ;
- le code applicatif à la logique JavaScript.

Les *aficionados* du développement Web auront probablement remarqué qu'une composante est absente du listing précédent : la présentation, généralement décrite via une feuille de style CSS. L'utilisation du CSS permet ainsi de définir un ensemble de styles applicables aux différentes balises de l'interface graphique. Android inclut un système proche des CSS : les styles et les thèmes.

Souvent sous-utilisés, les styles et thèmes Android permettent de grandement factoriser les styles et donc de conserver une cohérence à travers une application. C'est une des raisons pour lesquelles GreenDroid ([Lien 16](#)) utilise largement ces possibilités. Nous allons donc en faire une rapide présentation dans les lignes qui suivent.

Note :

L'ensemble des démos de cet article sont disponibles dans cette archive : [Lien 17](#).

Cette dernière ne contient qu'un unique projet avec lequel il vous faudra "jouer" pour appliquer telle ou telle méthode de style. N'hésitez pas à laisser un commentaire si vous n'arrivez vraiment pas à obtenir un résultat fonctionnel !

2. Appliquer un style à une balise

Prenons le code suivant :

```
<?xml version="1.0" encoding="utf-8"?>
<TextView
    xmlns:android="http://schemas.android.com/apk
/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
```

```
    android:gravity="center"
    android:text="Bonjour !!!" />
```

Ce code n'a rien de compliqué et est même plutôt "classique". Par défaut Android oblige le développeur à définir une hauteur et une largeur pour chacune des vues. C'est ce qui nous oblige à souvent ajouter du code de style `android:layout_[width|height]="[wrap_content|fill_parent]"`. Il faut avouer que ça peut devenir assez rébarbatif. Il est néanmoins possible de réduire le code de la façon suivante :

```
<TextView
    xmlns:android="http://schemas.android.com/apk
/res/android"
    style="@style/FillParent"
    android:gravity="center"
    android:text="Bonjour !!!" />
```

Note : L'attribut XML `style` n'appartenant pas au namespace android, il n'est pas nécessaire de le préfixer par android.

Pour ce faire, il suffit de créer un style `FillParent` dans les styles du projet :

```
<style name="FillParent">
    <item
name="android:layout_width">fill_parent</item>
    <item
name="android:layout_height">fill_parent</item>
</style>
```

Note : On utilise généralement le fichier `res/values/styles.xml` pour définir l'ensemble des styles d'un projet. Sachez néanmoins que cela n'a rien d'obligatoire. La seule nécessité est de définir le style dans la balise `<resources />` dans un fichier présent dans `res/values`.

3. Notion d'héritage

Contrairement aux feuilles de style CSS, les styles Android ne fonctionnent pas en cascade. Ainsi une balise `<TextView>` contenue dans un `<LinearLayout>` disposant d'un style `@style/MyStyle` n'hérite aucunement des propriétés de son parent. Il est néanmoins possible de faire hériter un style d'un autre style. Si nous souhaitons maintenant créer un style forçant la vue à prendre la taille de son parent et à centrer son contenu nous pouvons définir :

```
<style name="FillParent.Centered"
```

```
parent="@style/FillParent">
    <item name="android:gravity">center</item>
</style>
```

Il est également possible de simplifier cette déclaration de la façon suivante :

```
<style name="FillParent.Centered">
    <item name="android:gravity">center</item>
</style>
```

Lorsque le nom d'un style est du format <Part1>.<Part2>, Android considère automatiquement que le parent est le style de nom @style/Part1.

En utilisant ce nouveau style, il est possible de simplifier notre layout exemple :

```
<TextView
    style="@style/FillParent.Centered"
    android:text="Bonjour !!!" />
```

4. Les thèmes

Les thèmes Android ne sont rien d'autre que des styles s'appliquant à l'ensemble des balises de vos layouts. Pour contrer le problème de android:layout_[width|height] obligatoires, on peut donc, par exemple, imaginer un thème initialisant l'ensemble de ces valeurs à fill_parent. On commence donc par créer un style nommé Theme.Custom héritant du thème Android Theme.Black. Traditionnellement, bien qu'un style soit un thème, on le place dans un fichier nommé res/values/themes.xml.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="Theme.Custom"
parent="@android:style/Theme.Black">
        <item
name="android:layout_height">fill_parent</item>
        <item
name="android:layout_width">fill_parent</item>
    </style>
</resources>
```

On applique ensuite ce thème à l'ensemble de l'application en ajoutant l'attribut XML android:theme="@style/Theme.Custom" à la balise <application /> de notre AndroidManifest.xml. Le code du layout est donc de la forme suivante :

```
<?xml version="1.0" encoding="utf-8"?>
<TextView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:gravity="center"
    android:text="Bonjour !" />
```

5. Les attributs de style

Utilisés tels quels, les thèmes n'ont que peu d'intérêt. En effet, appliquer une valeur par défaut à l'ensemble des balises de votre application est quelque peu hasardeux et inutile. Les thèmes prennent tout leur sens lorsqu'ils sont utilisés en conjonction avec les attributs de style.

Un attribut de style peut-être utilisé comme "indirection de

style". Plutôt que de longs discours, appliquons un attribut de style à notre <TextView />. On commence par créer un nouvel attribut nommé myTextViewAttr dans le fichier res/values/attrs.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <attr name="myTextViewAttr" />
</resources>
```

Il ne nous reste plus maintenant qu'à appliquer ce style au layout voulu :

```
<?xml version="1.0" encoding="utf-8"?>
<TextView
    xmlns:android="http://schemas.android.com/apk/res/android"
    style="?attr/myTextViewAttr"
    android:text="Bonjour !" />
```

La <TextView /> précédente dispose de l'attribut de style ? attr/myTextViewAttr. On notera bien la présence d'un caractère "?" en lieu et place du traditionnel "@". L'utilisation du point d'interrogation permet d'informer le système que la ressource de style recherchée n'est pas myTextViewAttr (nous aurions alors écrit @style/myTextViewAttr) mais un style "pointé" par l'attribut myTextViewAttr.

Si on lance notre application dans l'état, nous obtenons une erreur affirmant qu'aucun des attributs android:layout_[height|width] n'a été initialisé. Ce résultat est tout à fait logique car le style qui se cache derrière l'attribut myTextViewAttr n'a pas été défini. Pour le définir, il suffit de l'initialiser dans le thème de notre application !

```
<style name="Theme.Custom_2"
parent="@android:style/Theme.Black">
    <item
name="myTextViewAttr">@style/FillParent.Centered</item>
</style>
```

L'application est maintenant parfaitement fonctionnelle et applique le style @style/FillParent.Centered à l'ensemble des balises disposant de l'attribut de style ? attr/myTextViewAttr. Simple et puissant n'est-ce pas ?

6. Etendre des thèmes Android

L'ensemble des fonctionnalités décrites dans cet article sont accessibles au développeur aussi bien pour créer ses propres styles que pour étendre et modifier les thèmes existants d'Android. Ainsi, lorsque vous souhaitez développer une application à dominante "noir sur blanc", il suffit de créer un thème héritant de @android:style/Theme.Light et de modifier les attributs à votre convenance. Voici un exemple de thème Android supprimant la fameuse ombre en haut de l'écran et modifiant le style par défaut de la titlebar :

```
<style name="Theme.Custom_3"
parent="@style/Theme.Custom_2">
    <item
name="android:windowContentOverlay">@null</item>
    <item
```

```
name="android:windowTitleStyle">@style/WindowTitl
e</item>
    <item
name="android:windowTitleBackgroundStyle">@style/
WindowTitleBackground</item>
</style>
```

Les styles associés sont définis de la façon suivante :

```
<style name="WindowTitleBackground">
    <item
name="android:background">@drawable/title_bar</it
em>
</style>

<style name="WindowTitle"
parent="@android:style/WindowTitle">
    <item
name="android:textAppearance">@style/TextAppearan
ce.WindowTitle</item>
</style>

<style name="TextAppearance.WindowTitle"
parent="@android:style/TextAppearance.WindowTitle
">
    <item name="android:textStyle">bold|
italic</item>
</style>
```

Le résultat est quelque peu coloré !



A mon grand regret, la documentation sur les différents attributs de style d'Android est assez pauvre. Jetez donc un oeil à la documentation de R.attr ([Lien 18](#)) et n'hésitez surtout pas à regarder le code source de la plateforme pour connaître l'utilité de tel ou tel attribut. Stylez/Thémez bien !

Retrouvez l'article de Cyril Mottier en ligne : [Lien 19](#)

Interview de David Syer - Spring Batch Technical Lead

La rédaction Java a eu l'opportunité de poser quelques questions à David Syer, le technical lead sur Spring Batch, à l'occasion d'une présentation sur la programmation concurrente et les applications distribuées avec Spring ([Lien 20](#)) qu'il a donnée dans les locaux de la société Zenika, à Paris.

1. Présentation générale

1.1. David Syer, pouvez-vous vous présenter et nous expliquer un peu votre parcours chez SpringSource ?

Je vis à Londres. Je travaille pour SpringSource ([Lien 21](#)) depuis le début, quand la société s'appelait d'abord Interface21 ([Lien 22](#)). Puis elle a été renommée SpringSource en 2007. J'ai rejoint la société en 2006. Je pense que j'étais le numéro 24 dans la société, ou quelque chose comme ça.

À cette époque c'était une petite entreprise, tout le monde faisait tout. Un peu de développement, un peu de consulting, un peu de formation et le plus gros de nos revenus provenait des services, de la formation et du consulting. Et puis nous avons beaucoup grossi entre 2006 et 2009. Puis nous avons été rachetés par VMware ([Lien 23](#)) l'année dernière donc nous sommes maintenant une entreprise beaucoup plus grande.

Et depuis ce temps, je suppose, mon rôle a changé un peu parce que je fais moins de services et je fais plus de développements open-source. Donc vraiment ma principale responsabilité maintenant est la communauté open-source Spring : Spring Batch ([Lien 24](#)), Spring Integration ([Lien 25](#)), les projets sur lesquels je travaille. Je pense que je travaille de manière égale sur Spring Batch, Spring Integration, Spring AMQP ([Lien 26](#)). Et il y a un autre projet qui s'appelle Spring Batch Admin ([Lien 27](#)). Ce sont en gros mes quatre principaux projets.

Je travaille aussi pour le framework Spring Core. Je pense que cela représente 80 % de ce que je fais. Uniquement des activités de développeur et pour la communauté. Et j'aime faire un peu de consulting. Donc j'aime travailler pour des clients parce qu'on peut ainsi connaître les problèmes réels et cela permet en fait d'améliorer le produit.

2. Spring Batch

2.1. Comment vous est venue l'idée de créer Spring Batch ?

Essentiellement des clients. Vous savez, quand on est en clientèle, on voit beaucoup de personnes qui écrivent des frameworks de batch en Java parce qu'il n'y en avait pas vraiment sur le marché de l'open-source. Et aujourd'hui il en existe encore très peu. Dans tous les cas, il y avait peu

d'applications en 2006 qui permettaient d'aider dans la conception d'applications batchs. Donc vraiment elle est venue des clients.

2.2. Quelles sont les fonctionnalités apportées par la dernière version et quelles sont les évolutions futures ?

La version actuelle est la 2.1. Il y avait un gros changement entre la 1.0 et la 2.0. Il y a un changement moins important entre la 2.0 et la 2.1.

La 2.1 a ajouté de nouvelles facilités pour la configuration. Elle améliore aussi la scalabilité et la parallélisation. Donc les caractéristiques de base étaient déjà présentes en 2.0, nous les avons beaucoup améliorées dans la 2.1.

Une petite partie du futur développement va aller dans le projet Spring Batch Admin ([Lien 27](#)). C'est une sorte de runtime. En effet Spring Batch est un framework de programmation dans lequel vous écrivez une application pour un traitement de batch. Si vous avez ensuite besoin de la déployer dans un runtime, de la manager et la monitorer, des choses comme ça, alors Spring Batch Admin est l'outil fait pour ça. Vous pouvez imaginer ça comme un service ou une interface utilisateur graphique et extensible.

Donc il y a une interface graphique Web et un ensemble d'API et services qui peuvent être utilisés pour composer ses propres services ou ses propres interfaces utilisateurs. C'est là où vous avez des choses comme l'historique de l'exécution des jobs, l'arrêt, le démarrage, la vue de la progression (50 %, 60 %, 100 % terminés). C'est donc pour le management et le monitoring de l'exécution de batchs, en cours et passés. En gros comme la console d'une application. C'est écrit pour être extensible et embarqué. Le design est très léger et on peut l'étendre de plusieurs façons.

C'est donc un domaine pour lequel il y aura probablement une version 3.0 de Spring Batch à un certain point mais je n'ai pas encore commencé à y travailler. Ce sera probablement l'année prochaine. La raison pour laquelle je veux passer en 3.0 est que certaines des API ont été soigneusement conçues mais elles ne sont pas très bien adaptées à un déploiement plus orienté services. Donc nous allons peut-être jeter un coup d'œil à certaines APIs de bas niveau et les réorganiser.

2.3. Est-il prévu un support de Spring Batch dans Spring Roo ?

Oui, définitivement. Et le shell de Spring Roo est très intéressant. C'est un très bon environnement pour exécuter des choses de façon interactive. On pourra exécuter un job depuis un shell Spring Roo et ce sera très intéressant. J'ai juste besoin de temps ou de quelqu'un pour travailler dessus. Si quelqu'un veut contribuer.

Je pense que Roo a une belle architecture pour créer des add-ons et les publier dans la communauté ces jours-ci. C'était une nouveauté de Spring Roo 1.1 et il y a en fait un framework pour écrire un plug-in et le publier sur Internet de sorte que n'importe qui peut l'utiliser. Je pense que ce sera une grande contribution à la communauté en vérité : un plug-in Spring Roo pour Spring Batch.

2.4. Quel livre sur Spring Batch pouvez-vous nous conseiller de lire ?

À ma connaissance il n'existe qu'un seul livre, "Spring Batch in Action" et il n'a pas encore été publié.

Il est cependant disponible en preview (MEAP) : [Lien 28](#). Apparemment il marche bien en preview et comblera un trou sur le marché. Je ne fais pas partie des auteurs même si j'ai été en relation avec l'un d'entre eux, Arnaud Cogoluègnes ([Lien 29](#)).

3. Autres projets

3.1. Travaillez-vous sur d'autres projets pour SpringSource ? Je sais que vous travaillez pour Activiti.

Activiti n'est pas un projet SpringSource mais je suis un committer. Je travaille principalement sur Spring Integration et Spring AMQP (Advanced Message Queueing Protocol). Spring AMQP est un graphique de haut niveau que nous sommes en train de mettre en place sur AMQP.

En fait c'est un protocole de messagerie que nous sommes en train de wrapper, principalement basé sur Rabbit (un produit d'un projet SpringSource).

En principe, Spring AMQP peut être utilisé avec d'autres brokers AMQP mais Rabbit est celui sur lequel nous sommes en train de nous concentrer pour la draft release.

3.2. Quel est votre rôle dans Activiti ?

Mon rôle dans Activiti est en gros de veiller à ce qu'Activiti satisfasse les utilisateurs et les besoins de la communauté Spring. En effet les développeurs de la communauté open-source Spring sont très intéressés par le workflow et ils aiment faire les choses d'une certaine façon : injection de dépendance, annotations avec peut-être un peu de configuration XML.

Donc en gros j'aide l'équipe Activiti à écrire leur code d'une façon qui soit utilisable, consommable par Spring.

4. SpringSource

4.1. Vous qui avez connu Interface21 ainsi que SpringSource, est-ce que l'acquisition de cette dernière par VMware a intrinsèquement changé la philosophie de la société ? Est-elle toujours aussi indépendante ou est-ce que VMware a pris le contrôle ?

Nous faisons partie de VMware, nous prenons définitivement ceci en compte dans nos buts et nos objectifs sont conformes à ceux de la grande entreprise VMware. C'est une période très intéressante en vérité. En ce qui concerne les opérations de tous les jours, je pense que les ingénieurs SpringSource, les gens comme moi, n'ont pas été très affectés par ça. VMware nous a achetés parce qu'il voulait la communauté SpringSource de façon à être capable de garder l'ingénierie SpringSource en place.

Et donc les choses qui ont changé sont l'administration, les ventes, le marketing. Ces choses ont été très fortement affectées, ainsi que les ressources humaines. Et la grande entreprise nous influence sur la façon stratégique dont nous pensons nos nouveaux produits et planifions où nous mettons nos ressources parce que le cloud est très important pour VMware et la virtualisation évidemment. Et ces choses-là sont très intéressantes pour nous tous à SpringSource parce qu'elles sont aussi nouvelles et donc elles nous influencent et, espérons-le, nous les influençons aussi.

Nous faisons partie d'une plus grande entreprise. Mais nous sommes une division avec sa propre marque donc la marque SpringSource est encore active et je ne pense pas qu'elle disparaîtra parce qu'elle est très forte. Et donc la communauté open-source ne changera pas.

Donc vous êtes optimiste ?

Oui, ça a été une chose très positive en vérité.

5. Conclusion

5.1. Quel est votre ressenti sur l'avenir du langage Java ?

Je pense que c'est plutôt un vaste sujet. Premièrement je ne pense pas qu'il va disparaître parce qu'il existe un écosystème très riche et une grosse remontée d'expérience. Les JVM modernes ont été tellement améliorées que personne ne peut dire que Java n'est pas efficace. Et la récente montée en popularité d'autres langages de JVM tels que Groovy et Scala est très significative.

Mais il y a également une énorme montée en popularité et utilisation d'autres plateformes et langages (Ruby, Python, JavaScript) et nous devons en être conscients.

Peut-être que nous devrions tous prendre la résolution de Nouvel An d'apprendre et d'utiliser un autre langage et essayer d'évaluer quand il est plus adapté ?

5.2. Vous avez donné une présentation intéressante sur la programmation concurrente et les applications distribuées avec Spring, dans les bureaux de Zenika à Paris. Allez-vous revenir en France en 2011 pour donner d'autres présentations et formations ?

Je serai de nouveau à Paris en mars pour coanimer une formation sur Enterprise Integration ([Lien 30](#)), chez Zenika, avec Arnaud Cogoluègues, coauteur de "Spring Batch in Action".

Il est également prévu que je fasse quelques interventions en clientèle dans et autour de Paris, même si souvent ces interventions peuvent être traitées par nos consultants déjà

sur place.

6. Liens

- Zenika : [Lien 31](#)
- Spring Batch : [Lien 24](#)
- Spring Integration : [Lien 25](#)
- L'équipe d'Activiti BPM : [Lien 32](#)
- Slides de la présentation : [Lien 33](#)
- Le code source des démos : [Lien 34](#)

Retrouvez l'article de Celinio Fernandes et Gildas Cuisinier en ligne : [Lien 35](#)

Zend Framework

Les derniers tutoriels et articles



Installer le framework Zend et créer un projet ZF en 15 minutes max

Vous verrez dans cet article comment installer le framework Zend et créer un projet ZF, le tout en une quinzaine de minutes.

1. Avant de démarrer l'installation

Tout d'abord, il faut télécharger la version la plus récente à cette adresse : [Lien 36](#)

Il faudra peut-être s'enregistrer sur le site pour pouvoir le télécharger ; dans ce tutoriel j'ai utilisé la version 1.10.7, mais il vaut mieux télécharger la version 1.11 qui vient juste d'être mise en ligne.

Product	Version	Format/Size	Notes
Zend Framework 1.11 full (recommended)	1.11.0	(ZIP) 25.49 MB	Learn more

Si un serveur HTTP est installé sur votre machine, vous pouvez passer directement à l'étape Installation du framework page 5, sinon suivez les étapes pour installer WAMP, qui est très simple à installer et à utiliser.

Téléchargez ensuite la dernière version de WAMP via l'une des adresses ci-dessous :

- site officiel : [Lien 37](#) ;
- lien direct : [Lien 38](#) (la version actuelle à l'heure d'écriture du tutoriel).

ATTENTION : N'installez pas WampServer 2 par dessus WAMP5. Si vous avez une version de WAMP5 installée, sauvegardez vos données, désinstallez la, et supprimez le répertoire de WAMP5 avant d'installer WampServer 2.



Installez ensuite WAMP, en suivant les instructions, dans le répertoire par défaut : « C:\wamp », cela devrait installer au minimum la version 5.3.0 de PHP, la version 2.2.11 d'Apache et la version 5.1.36 de MySQL.

Pour vérifier que WAMP s'est bien installé, faites un double-clic sur le raccourci qui vient d'être créé sur votre Bureau :



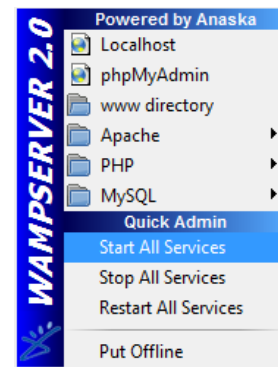
WAMP va se lancer en tâche de fond et une icône devrait apparaître en bas à droite du Bureau dans la barre des

tâches :

L'icône passera par les trois stades suivants :

- aucun service n'est démarré ;
- un service est démarré ;
- tous les services sont démarrés, WAMP est prêt.

Si vous n'avez pas la troisième icône, essayez un clic gauche sur l'icône, puis un clic sur « Start all services ».



Si l'icône n'a toujours pas changé et que vous ne voyez pas « Put Offline » mais « Put Online », faites un clic sur « Put Online ».

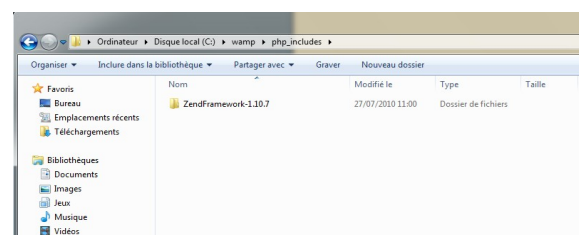
Si ça ne fonctionne toujours pas, je vous invite à aller voir sur le forum du site officiel : Forum officiel ([Lien 39](#)).

Une fois que vous avez cette icône , vous pouvez commencer l'installation du framework Zend.

2. Installation du framework

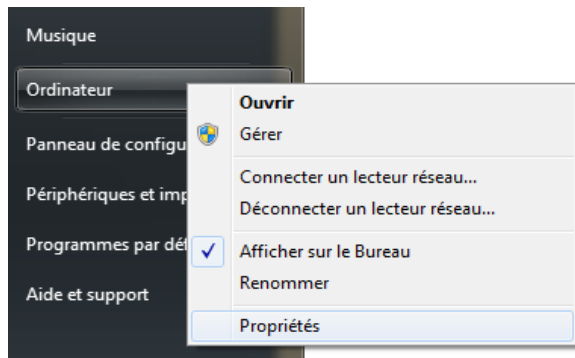
Une fois le framework téléchargé (cf. page 3), décompressez-le dans le répertoire : C:\wamp\php_includes.

Si ce dossier n'existe pas, créez-le (clic droit puis Nouveau Dossier). Après l'avoir décompressé, vous devriez avoir :

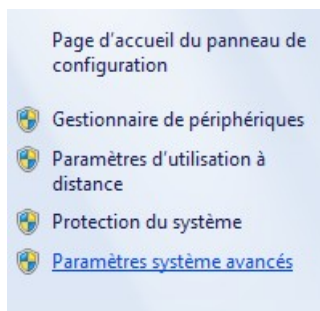


2.1. Sous Windows 7

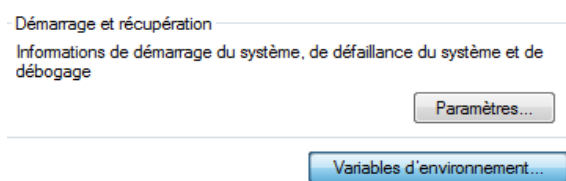
Allez dans Démarrer, faites un clic droit sur Ordinateur, puis cliquez sur propriétés.



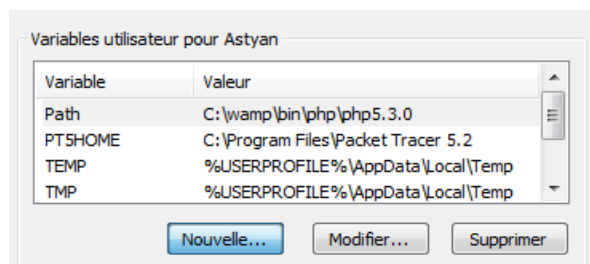
Cliquez ensuite sur Paramètres système avancés en haut à gauche :



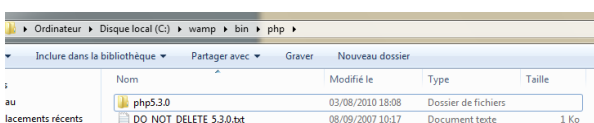
Cliquez ensuite sur « Variables d'environnement » :



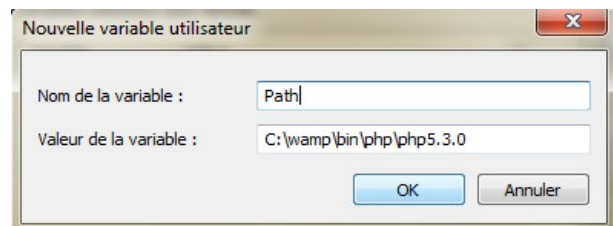
Cliquez sur « Nouvelle » :



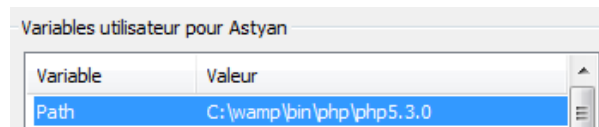
Si vous n'avez pas installé WAMP 2.0i, allez voir dans **C:\wamp\bin\php** les dossiers qui existent et notez le nom du dossier PHP suivi du nombre le plus élevé (ici php5.3.0).



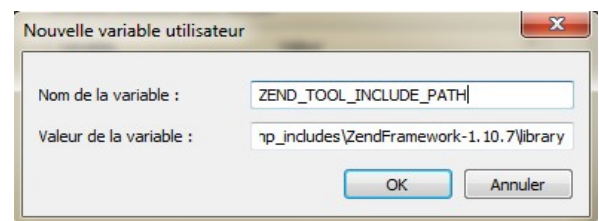
Sinon, tapez ceci, puis cliquez sur OK :



Vous devriez voir :



Refaites ensuite la même manipulation pour créer la variable suivante :



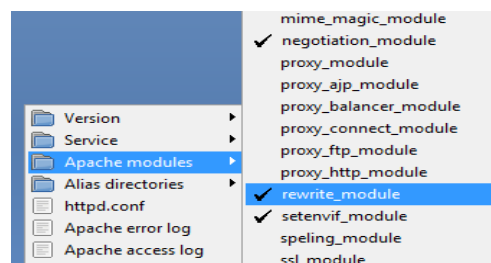
La valeur de la variable est la suivante : **C:\wamp\php_includes\Zendframework-1.10.7\library** à modifier selon votre version du framework (changez juste le numéro).

Cliquez enfin sur OK pour appliquer les changements.

Pour pouvoir utiliser Zend Tools (pour créer le projet Zend), il faut aller dans **C:\wamp\php_includes\Zendframework-1.10.7\bin** puis copier les deux fichiers suivants : « zf.bat » et « zf.php ».

Allez ensuite dans **C:\wamp\bin\php\php5.3.0** puis collez vos deux fichiers dedans.

Un dernier petit réglage, il faut activer le module « rewrite » d'Apache, en cliquant sur l'icône de WAMP, puis en allant dans Apache puis Apache Modules, suivi d'un clic sur la ligne « rewrite_module ».



Voilà, le framework Zend est installé !

Normalement cela devrait fonctionner sans redémarrer, mais dans le doute faites un redémarrage.

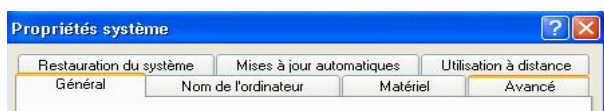
Une fois votre machine redémarrée, on peut passer à la création d'un projet Zend, passez directement au chapitre 3.

2.2. Sous Windows XP

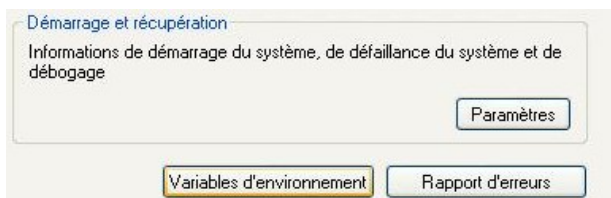
Faites un clic droit sur Poste de Travail, puis cliquez sur Propriétés :



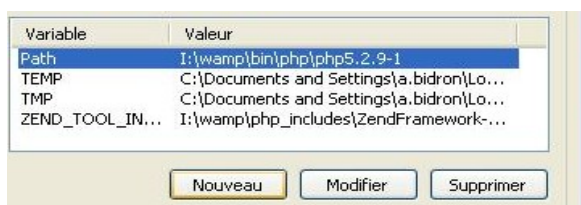
Allez ensuite dans l'onglet « Avancé » :



Puis allez dans « Variables d'environnement » :



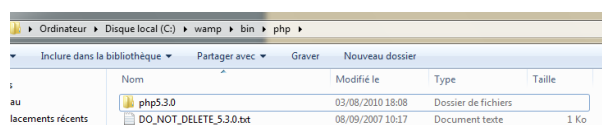
Cliquez sur « Nouveau ».



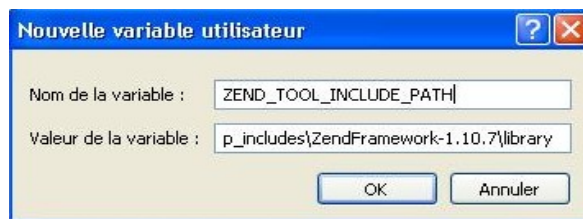
Tapez la même chose si vous avez installé WAMP 2.0i :



Sinon, allez voir dans **C:\wamp\bin\php** les dossiers qui existent, et notez le nom du dossier php suivi du nombre le plus élevé (ici php5.3.0).



Refaites ensuite la même manipulation pour créer la variable suivante :



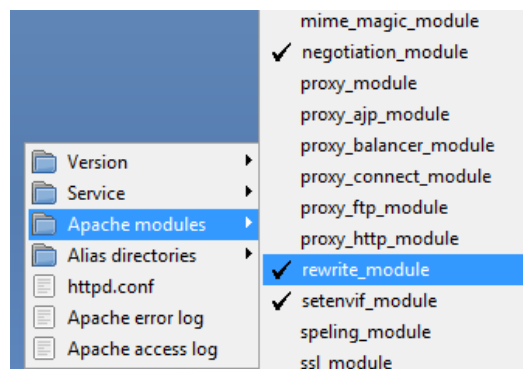
La valeur de la variable est la suivante : **C:\wamp\php_includes\Zendframework-1.10.7\library** à modifier selon votre version du framework (changez juste le numéro).

Cliquez enfin sur OK pour appliquer les changements.

Pour pouvoir utiliser Zend Tools (pour créer le projet Zend), il faut aller dans **C:\wamp\php_includes\Zendframework-1.10.7\bin** puis copier les deux fichiers suivants : « zf.bat » et « zf.php ».

Allez ensuite dans **C:\wamp\bin\php\php5.3.0** puis collez vos deux fichiers dedans.

Un dernier petit réglage, il faut activer le module « rewrite » d'Apache, en cliquant sur l'icône de WAMP, puis en allant dans Apache puis Apache Modules, suivi d'un clic sur la ligne « rewrite_module ».



Voilà, le framework Zend est installé !

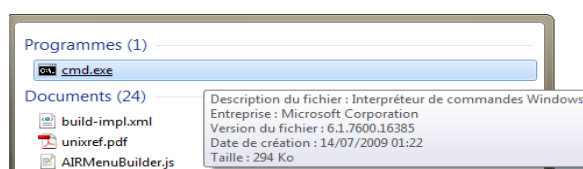
Normalement cela devrait fonctionner sans redémarrer, mais dans le doute faites un redémarrage.

Une fois votre machine redémarrée, on peut passer à la création d'un projet Zend, page suivante !

3. Création d'un projet Zend

Pour vérifier que le framework Zend se soit bien installé, allez dans démarrage,

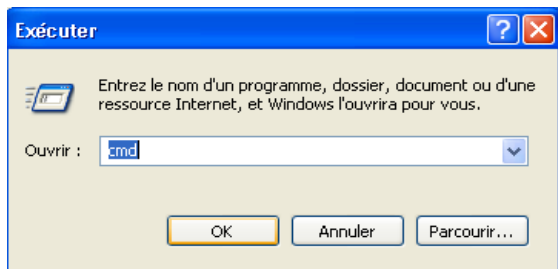
- Sous Windows 7 tapez « cmd » dans la barre de recherche, puis appuyez sur la touche Entrée, ou cliquez sur la ligne cmd.exe :



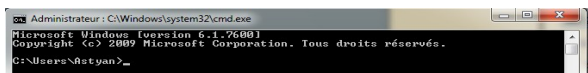
- Sous Windows XP, cliquez sur « Exécuter » :



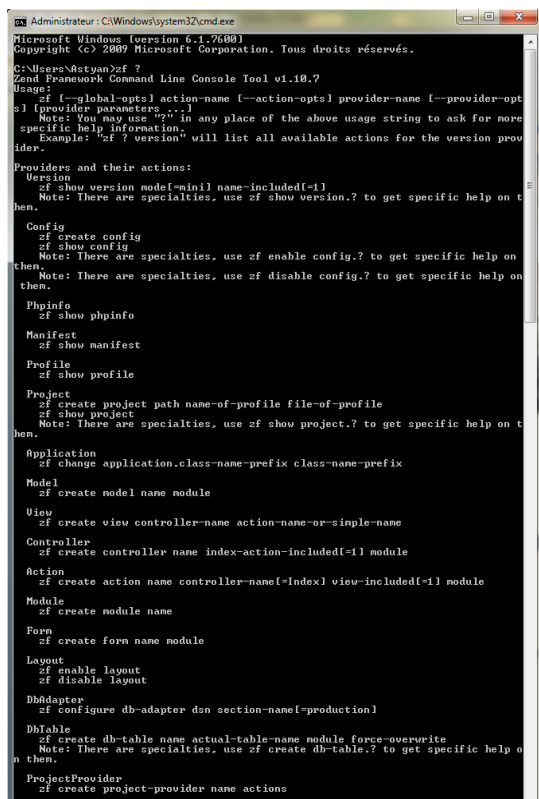
Puis tapez « cmd » et appuyez sur OK :



Vous devriez voir cette fenêtre :



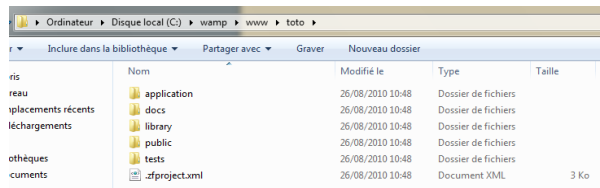
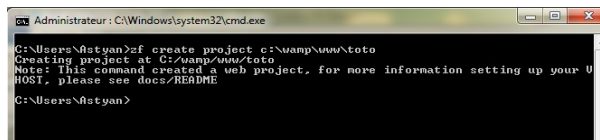
Tapez maintenant « zf ? », vous devriez voir cet écran si l'installation a fonctionné :



Si vous avez un message d'erreur, tapez-le sur Google, vous devriez vite trouver une solution. Certains problèmes étant très fréquents, en suivant pas à pas ce tutoriel, vous ne devriez pas avoir de message d'erreur, mais tout peut arriver.

Tapez ensuite la commande magique : « **zf create project**

c:\wamp\www\NomDuProjet » ; vous devez voir ceci :



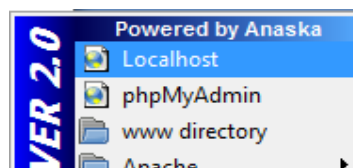
Voilà, vous avez créé un projet nommé toto dans le répertoire de WAMP, il ne reste plus qu'à rajouter les bibliothèques de Zend pour pouvoir enfin coder.

Pour cela, allez dans **C:\wamp\php_includes\Zendframework-1.10.7\library** et copiez le dossier « Zend ».

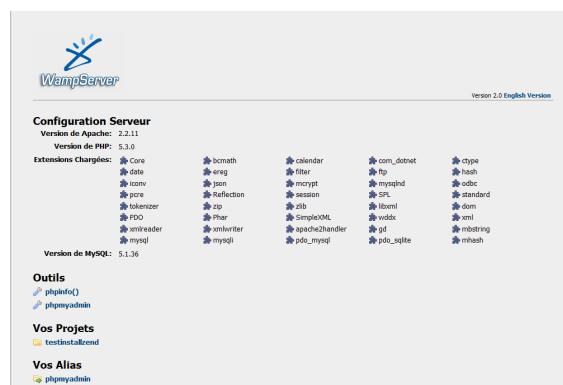
Allez ensuite dans **C:\wamp\www\toto\library** et collez votre dossier « Zend ».

Testons maintenant tout cela en affichant la page par défaut d'un projet ZF.

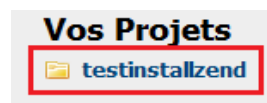
Cliquez sur l'icône de WAMP dans la barre des tâches, puis cliquez sur « Localhost ».



Vous devriez arriver sur cette page :



Supposons que vous ayez appelé votre projet "testinstallzend", cliquez dessus.



Vous devriez voir ceci :

Index of /testinstallzend

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 application/	12-Dec-2010 12:38	-	
 docs/	12-Dec-2010 12:38	-	
 library/	12-Dec-2010 12:43	-	
 public/	12-Dec-2010 12:38	-	
 tests/	12-Dec-2010 12:38	-	

Cliquez maintenant sur « public ».

Vous devez voir la page par défaut de tout projet ZF :



Votre projet est maintenant fin prêt, il ne vous reste plus qu'à coder avec votre IDE préféré, ou un éditeur de texte.

Retrouvez l'article de Michael Vaudin en ligne : [Lien 40](#)

Utiliser le moteur de rapports de SQL Server en PHP

Comment intégrer le SDK PHP de Microsoft pour SSRS dans une application Zend Framework

Cet article explique comment afficher des rapports générés par SQL Server Reporting Services (SSRS) en PHP, en utilisant le SDK dédié mis gratuitement à disposition par Microsoft, dans une application PHP Zend Framework.

1. Présentation de SSRS et de son SDK PHP

Au cas où vous n'êtes pas familier du terme, ce qu'on appelle un rapport (anciennement appelé "état") est un document de synthèse de certaines informations d'une base de données métier, présenté en général sous forme de tableaux et de graphiques, qui a pour vocation d'aider à la prise de décisions. Ce qui explique que cela fait partie du monde de la Business Intelligence (BI). Le "reporting" désigne l'action de créer et diffuser des rapports. Mais beaucoup d'entre vous font déjà du reporting sans le savoir. Afficher une liste des clients ou des dernières ventes d'un commerce en PHP, que ce soit dans une "simple" page html ou en PDF, c'est un rapport. Alors pourquoi des outils dédiés ? Il y a plusieurs raisons à cela, parmi lesquelles :

- l'éditeur de rapports permet de générer des documents à la mise en page complexe et agréable à l'œil avec un outil visuel (accessible aux non-informaticiens, ce qui est très intéressant car cela permet à certains utilisateurs de créer eux-mêmes leurs rapports selon leurs besoins et leurs préférences de travail) ;
- les données contenues dans le rapport peuvent être agrégées depuis de multiples sources : bases de données relationnelles classiques, datawarehouse (entrepôt de données dédié aux analyses BI), fichiers plats dans de multiples formats, webservices, et même des classes de vos propres applications ;
- le moteur de reporting permet d'effectuer le rendu d'un rapport unique dans de multiples formats nativement : HTML, PDF, Word, Excel... ;
- le serveur de rapports permet de paramétrer finement l'accès aux rapports et aux données qu'ils contiennent.

SQL Server Reporting Services (SSRS) est la solution de reporting de Microsoft pour sa base de données SQL Server. La solution se compose de plusieurs outils :

- deux éditeurs de rapports (Report Builder, plutôt destiné à l'utilisateur final, ou BI Development Studio, plutôt dédié au professionnel BI) ;
- un backoffice permettant de diffuser les rapports de manière sécurisée grâce à une politique de droits d'accès poussée ;
- un webservice SOAP pour rendre les rapports accessibles depuis des applications tierces.

C'est donc le webservice qui va nous intéresser ici. On pourrait y accéder directement en PHP grâce à la classe standard SoapClient ou à son équivalent Zend_Soap_Client, mais Microsoft nous rend la tâche encore plus facile en mettant à disposition un SDK qui est un ensemble de classes PHP de haut niveau encapsulant les appels SOAP. Le code PHP pour récupérer un rapport est ainsi minimal.

Existe-t-il un ou plusieurs équivalents à SSRS en libre ?

Oui bien sûr ! Le plus célèbre étant JasperServer ([Lien 41](#)). Je vous invite d'ailleurs à lire le livre blanc sur le monde de la BI open source pour plus d'informations ([Lien 42](#)).

2. Intégration du SDK dans une application Zend Framework

2.1. Installation du SDK et configuration de SSRS

La première étape sera bien sûr de télécharger le SDK ([Lien 43](#)). Après extraction de l'archive, seul le répertoire

"bin" est à intégrer dans notre projet PHP, le reste étant de la documentation. Nous partons d'une application MVC avec l'arborescence standard de ZF (version 1.8 ou supérieure) telle que décrite dans la documentation officielle ([Lien 44](#)). Nous allons donc créer un répertoire **library/SSRS** et y copier le contenu du répertoire "bin" du SDK SSRS.

Il va falloir ensuite créer un accès pour notre application et pour cela SSRS utilise l'authentification Windows. Il faut donc ajouter un compte utilisateur Windows (ou utiliser le vôtre à des fins de test), et autoriser ce compte à accéder à la base de données. Pour cela, rendez-vous dans l'outil d'administration de SQL Server, allez dans les propriétés de sécurité / connexion, ajoutez l'utilisateur (avec le nom complet MACHINE\username ou DOMAIN\username). Sélectionnez ensuite la base de données qui servira aux rapports, ajoutez le compte utilisateur et donnez-lui les privilèges **db_datareader** et **db_datawriter**.

Reste maintenant à ajuster le degré de sécurité de SSRS. Pour cela, ouvrez le fichier **rsreportserver.config** localisé dans le répertoire de SSRS : C:\Program Files\Microsoft SQL Server \MSRS10.SQLEXPRESS\Reporting Services\ReportServer. Ce fichier est au format XML, et il faut ajouter l'élément **<RSWindowsBasic/>** au noeud **<AuthenticationTypes>** :

```
<Authentication>
  <AuthenticationTypes>
    <RSWindowsNegotiate/>
    <RSWindowsNTLM/>
    <RSWindowsKerberos />
    <RSWindowsBasic/>
  </AuthenticationTypes>
  <EnableAuthPersistence>true</EnableAuthPersistence>
</Authentication>
```

La directive **RSWindowsNegotiate** permet de basculer en Kerberos ou NTLM selon le protocole d'authentification supporté.

2.2. Réalisation d'un helper d'action

Pour pouvoir récupérer des rapports dans toute l'application, le plus simple est de créer un helper d'action ([Lien 45](#)) qui pourra donc être appelé depuis n'importe quelle action de contrôleur.

La première chose à faire est de stocker les paramètres de connexion : URL du webservice, identifiant et mot de passe. Nous utilisons donc le fichier **application/configs/application.ini** dans lequel nous ajoutons les lignes suivantes :

```
ssrs.url = "http://localhost:85/ReportsServer"
ssrs.username = "MACHINE\toto"
ssrs.password = "xxx"
```

Attention ! J'ai constaté une bizarrerie avec le SDK SSRS : celui-ci semble utiliser en interne le nom de machine. Qu'on appelle le serveur de reporting par son IP ou par son nom, il est impératif de préciser dans le fichier hosts du serveur PHP le nom de la machine SSRS avec son IP. Pour localhost c'est bien sûr inutile car l'entrée existe déjà par

défaut dans le fichier hosts, mais si votre SSRS est sur une autre IP, il faudra le renseigner.

Mettez donc une ligne comme celle-ci dans votre fichier C:\Windows\system32\drivers\etc\hosts (ou /etc/hosts sous Linux) :

Fichier hosts

```
192.168.1.5    reportserver
```

Si ce n'est pas déjà fait, rendez la configuration accessible aisément dans toute votre application en ajoutant la méthode suivante au bootstrap :

application/Bootstrap.php

```
protected function _initConfig()
{
    $config = new Zend_Config($this->getOptions());
    Zend_Registry::set('config', $config);
    return $config;
}
```

La configuration sera ainsi accessible partout via un simple **Zend_Registry::get('config')**.

Nous créons ensuite le fichier **library/Application/Helper/Rapport.php**. La classe ne comportera qu'une méthode **direct()** ce qui permet au helper d'être appelé directement dans les contrôleurs (pour plus de détails sur la manière de réaliser ses propres helpers d'action, voyez le manuel de Zend Framework : [Lien 45](#)). Voici le code source :

```
class Application_Helper_Rapport extends Zend_Controller_Action_Helper_Abstract
{
    /**
     * Récupère un rapport SSRS et l'affiche inline au format PDF
     * @param array $options paramètres d'affichage
     */
    public function direct($options)
    {
        $controller = $this->getActionController();
        $config = Zend_Registry::get('config');
        require_once(ROOT_PATH . '/library/ssrs/SSRSReport.php');
        try {
            // se connecte au webservice de SSRS
            $ssrs = new SSRSReport(new Credentials($config->ssrs->user, $config->ssrs->password), $config->ssrs->url);
            // charge le rapport demandé
            $report = $ssrs->LoadReport2($options['rapport']);
            // passe les éventuels paramètres
            if (array_key_exists('param', $options) and is_array($options['param']) and count($options['param']) > 0) {
                $params = array();
                foreach ($options['param'] as $key => $val) {
                    $param = new ParameterValue();
                    $param->Name = $key;
                    $param->Value = $val;
                }
            }
        } catch (Exception $e) {
            // gestion des erreurs
        }
    }
}
```



```

        $params[] = $param;
    }
    $ssrs->
>SetExecutionParameters2($params);
    }
    // effectue le rendu en PDF
    $renderPdf = new RenderAsPDF();
    $rapport = $ssrs->Render2($renderPdf,
        PageCountModeEnum::$Actual,
        $Extension,
        $MimeType,
        $Encoding,
        $Warnings,
        $StreamIds);

    // prépare l'affichage en purgeant la
vue et en mettant les bons headers HTTP
    $controller->
>getHelper('viewRenderer')->setNoRender();
    $controller->getHelper('layout')->
>disableLayout();
    $controller->getResponse()-
>clearBody();
    $controller->getResponse()-
        ->setHeader('Content-
Disposition', 'inline; filename=document.pdf')
        ->setHeader('Content-
type', 'application/pdf');
    ;
    // affiche le contenu du rapport sur
le flux de sortie
    echo $rapport;
    } catch (SSRSReportException
$serviceException) {
        throw new
Exception($serviceException->GetErrorMessage());
    }
}
}

```

La méthode `Render2()` prend un certain nombre de paramètres par référence, qui contiendront différentes informations sur la production du rapport au retour de l'appel de fonction. Dans le cas présent, je n'en ai pas besoin, mais cela peut servir par exemple à récupérer les images jointes pour un rapport affiché au format HTML.

Nous avons ici traité uniquement le cas des rapports PDF, mais les méthodes sont sensiblement équivalentes pour les autres formats, nous vous invitons à consulter la documentation fournie avec le SDK. Il faudra éventuellement adapter les headers http, et surtout le type mime (`application/*`) si vous utilisez un autre format que le PDF comme illustré dans l'exemple.

Il faut enfin veiller à ce que notre helper soit chargé dans le Bootstrap, pour cela créez une méthode `_initHelpers`, ou ajoutez une ligne à votre méthode existante :

```

application/Bootstrap.php
protected function _initHelpers()
{
    Zend_Controller_Action_HelperBroker::addHelper(
        new Application_Helper_Rapport());
}

```

2.3. Récupération d'un rapport dans un contrôleur

La récupération d'un rapport consiste en un seul appel de fonction dans une action de contrôleur, comme illustré ci-dessous :

```

public function printAction()
{
    $this->_helper->rapport(
        'rapport' => "/chemin/vers/le/rapport",
        'param' => array(
            'param1' => 'valeur1',
            'param2' => 'valeur2',
        ),
    );
}

```

3. Conclusion

Comme vous avez pu le voir, l'affichage de rapports SSRS dans une application PHP est très simple grâce au SDK. Vous pourrez donc aisément utiliser du reporting professionnel dans vos applications web basées sur SQL Server.

Retrouvez l'article d'Olivier Van Hoof en ligne : [Lien 46](#)

Créer une image ronde en CSS3

Avec le CSS3, on peut faire des choses qui nécessitaient autrefois l'utilisation d'un logiciel de design (type Photoshop ou Gimp). Par exemple, comment rendre une image ronde ? Il existe deux méthodes : tout d'abord une technique qui ne fonctionne que sous Chrome et Safari, et ensuite une technique qui fonctionne sur tous les navigateurs modernes (Firefox, Opera, Safari, Chrome et IE9).

Compatibilité : Chrome, Safari, Firefox, Internet Explorer 9, Opera

1. Technique pour Chrome et Safari

Le principe est relativement simple, il suffit d'utiliser la propriété **border-radius** CSS3 (et ses équivalents pour les navigateurs : `-moz-border-radius`, `-webkit-border-radius`). Cette propriété est appliquée sur un élément qui contient une image. On lui applique également la propriété **overflow** avec pour valeur **hidden**.

Démo : [Lien 47](#)

```
<style type="text/css">
.roundedImage{
    overflow:hidden;
    -webkit-border-radius:50px;
    -moz-border-radius:50px;
    border-radius:50px;
    width:90px;
    height:90px;
}
</style>
<div class="roundedImage">
    
</div>
```

(y compris Safari et Chrome).

Démo : [Lien 48](#)

```
<style type="text/css">
.roundedImage{
    overflow:hidden;
    -webkit-border-radius:50px;
    -moz-border-radius:50px;
    border-radius:50px;
    width:90px;
    height:90px;
}
</style>
<div class="roundedImage"
style="background:url(/images/slideshow/1.jpg)
no-repeat 0px 0px;">
    &nbsp;
</div>
```

3. D'autres exemples

Démo : [Lien 49](#)

4. Conclusion

La technique la plus compatible reste encore de mettre l'image au niveau du div dans la propriété css background mais du point de vue référencement c'est un manque à gagner.

Retrouvez l'article de Jérôme Debray en ligne : [Lien 50](#)

2. Technique pour tous les navigateurs

Vous noterez tout de même que sous Firefox, Opera et Internet Explorer 9, la première technique ne fonctionne pas. Il y a donc une astuce, il suffit de ne pas utiliser de balises image et d'utiliser à la place la propriété **background** et cela fonctionnera sous tous les navigateurs

Multiples background en CSS3

Le CSS3 implémente une nouvelle fonctionnalité permettant d'ajouter plusieurs images en arrière-plan d'un conteneur (par exemple un div). Cette fonctionnalité permet d'augmenter grandement votre créativité en web design. Compatibilité : Chrome, Safari, Firefox, Internet Explorer 9, Opera.

1. Principe de fonctionnement

Le principe est très simple, il suffit d'ajouter autant d'URL d'images que nécessaire dans la propriété **background**.

Voici un exemple : [Lien 51](#)

```
.multipleBackground{
    position:relative;
    width:200px;
    height:120px;
```

```
background:url(..images/site/browsers/chrome.png) 0 0 no-repeat,
    url(..images/site/browsers/firefox.png) 30px 30px no-repeat,
    url(..images/site/browsers/safari.png) 60px 60px no-repeat,
    url(..images/site/template/projects.png) 0 0 no-repeat;
}
```

Vous remarquerez que l'image en premier plan est déclarée

en premier dans la propriété **background**.

2. Avec un dégradé

Il est à noter que l'on peut inclure un gradient (ne fonctionne pas sous Opera, ni sous IE9) :

demo : [Lien 52](#)

```
.multipleBackgroundGradient{
  position:relative;
  width:200px;
  height:120px;
  /*Pour Navigateur Webkit*/
  background:url(..images/site/browsers/chrome.png) 0 0 no-repeat,
  url(..images/site/browsers/firefox.png) 30px 30px no-repeat,
```

```
url(..images/site/browsers/safari.png) 60px 60px no-repeat,
  -webkit-gradient(linear, left top, left bottom, from(#68cbd0), to(#000000));
  /*Pour Navigateur Mozilla*/
  background:url(..images/site/browsers/chrome.png) 0 0 no-repeat,
  url(..images/site/browsers/firefox.png) 30px 30px no-repeat,
  url(..images/site/browsers/safari.png) 60px 60px no-repeat,
  -moz-linear-gradient(top center, #68cbd0, #000000);
}
```

le gradient étant en arrière-plan, il est déclaré en dernier.

Retrouvez l'article de Jérôme Debray en ligne : [Lien 53](#)

Les bordures en CSS3

Le CSS3 améliore la propriété border afin de lui attribuer plus de fonctionnalités.

NOTE : selon le navigateur, il est nécessaire d'ajouter ou non le préfixe pour utiliser la propriété CSS3 : -webkit, -moz, -o. Si c'est le cas, il sera mis dans les exemples de code ci-dessous.

Compatibilité : Chrome, Safari, Firefox, Internet Explorer 9, Opera.

1. border-color

Cette propriété permet de changer les couleurs des bordures.

border-color: top right bottom left;
ou **border-color: top|bottom right|left;**

```
.border-color{
  border:5px solid #eee;
  background-color: #fff;
  border-color:#900 transparent #900 transparent;
  width:200px;
  height:30px;
}
<div class="border-color">
  border-color
</div>
```

Démo : [Lien 54](#)

2. border-image (non compatible : IE9 , compatibilité partielle : Opera, Firefox > 3.6)

La bordure peut être dorénavant mise en forme via des **images**. On peut ainsi spécifier une ou des images pour chaque bordure (top, left, bottom, right) ainsi qu'une ou des images pour les coins des bordures via les propriétés suivantes :

- **border-image** ;
- **border-top-image** ;
- **border-right-image** ;
- **border-bottom-image** ;
- **border-left-image**.

```
.borderImage{
  width:200px;
  height:30px;
```

```
border: solid transparent;
border-width:12px 22px;
-webkit-border-image:
url(..images/site/template/puceTitle.png) 12 22 round;
-moz-border-image:
url(..images/site/template/puceTitle.png) 12 22;
border-image:
url(..images/site/template/puceTitle.png) 22 round;
}
```

Démo : [Lien 55](#)

La propriété **round** permet de répéter l'image (tout comme la propriété **repeat**). Si l'on préfère étendre l'image sur la bordure, on peut utiliser la valeur **stretch**.

```
.borderImageStretch{
  width:200px;
  height:30px;
  border: solid transparent;
  border-width:12px 22px;
  -webkit-border-image:
url(..images/site/template/puceTitle.png) 12 22 stretch;
  -moz-border-image:
url(..images/site/template/puceTitle.png) 12 22;
  border-image:
url(..images/site/template/puceTitle.png) 22 stretch;
}
```

Démo : [Lien 56](#)

3. border-radius

Permet de modifier l'**arrondi** des coins des bordures d'un élément.

On peut appliquer l'**arrondi** à un seul côté.

```
.border-radius{
  -moz-border-radius: 10px;
  -webkit-border-radius: 10px;
  border-radius:10px;
}

.border-radius-test2{
  -moz-border-radius:10px 0px 0px 0px;
  -webkit-border-radius:10px 0px 0px 0px;
  border-radius:10px 0px 0px 0px;
}
```

Démo : [Lien 57](#)

4. box-shadow

Le **box-shadow** permet de rajouter un ombrage à un

élément. Cet ombrage est lié au **border-radius**. En effet, si l'élément a un ou des coins arrondis, l'ombrage sera également arrondi dans le ou les coins.

```
.shadow{
  -moz-box-shadow: 5px 5px 5px #000000;
  -webkit-box-shadow: 5px 5px 5px #000000;
  -o-box-shadow: 5px 5px 5px #000000;
  box-shadow: 5px 5px 5px #000000;
}
```

Démo : [Lien 58](#)

Retrouvez l'article de Jérôme Debray en ligne : [Lien 59](#)

Développement Web

Les derniers tutoriels et articles

Tester ses navigateurs de façon fiable

Pour une fois, je ne parlerai pas d'HTML5, de JavaScript avancé ou de performances mais de bonnes pratiques. On reste toutefois dans la modernité du développement Web, car nous allons voir pourquoi j'estime que la plupart des développeurs Web et pire, des agences Web, sont sous-équipés lorsqu'il s'agit de tester leurs navigateurs.

Je vous livre même la conclusion immédiatement : vous devez avoir des machines virtuelles pour avoir un environnement de test simplement normal et voici pourquoi.

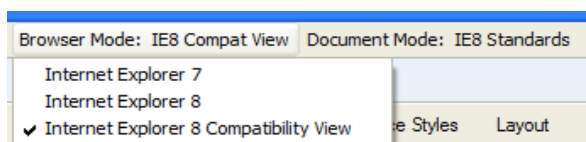
L'article original peut être lu sur le site BrainCracking ([Lien 60](#)) : Tester fiablement ses navigateurs ([Lien 61](#))

1. Je ne m'énerve pas, j'explique

Je travaille dans des conditions outrageusement luxueuses. Comprenez par là que je travaille sur un MacBook Pro depuis quatre ans et que j'ai une machine virtuelle pour chaque version d'Internet Explorer. J'ignorais ma condition de nant jusqu'au jour où j'ai travaillé avec une agence Web puis discuté avec d'autres développeurs.

Ce qui, comme la plupart des développeurs ne me gênait pas au début (après tout c'est cher à mettre en place) est devenu rapidement un problème : mes prestataires ne pouvaient pas reproduire certains bogues, car ils n'avaient pas les outils nécessaires. Après une petite année à ce régime, j'ai accumulé suffisamment de preuves pour me permettre de demander légitimement aux prochains prestataires de s'équiper avant de travailler avec nous.

2. Arrêtez les « compatibility mode »



C'est probablement le pire moyen de tester : le mode de compatibilité d'IE8 ou IE9 supposé exécuter le moteur de rendu d'IE7 ou IE8. Microsoft a voulu bien faire pour faciliter la vie de ses utilisateurs et leur permettre une transition en douceur de IE7 vers IE8, mais les développeurs se sont vite rendu compte que **le mode de compatibilité génère des rendus différents de IE7**. À tel point que Yahoo! et sa matrice de navigateurs à tester recommandent constamment de tester spécifiquement le mode de compatibilité (voir les updates, dans les notes ([Lien 62](#))).

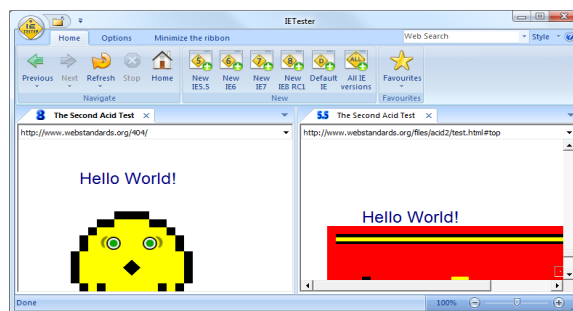
Certaines techniques sont faussées : j'ai, par exemple, constaté que le support des images en data:uri et base64 fonctionnait dans IE8 en mode IE7 standard alors que IE7 ne le supporte pas. Comment faire pour valider certaines techniques émergentes (ici pour les performances) si vous n'avez pas les bons navigateurs pour tester ?

La différence de comportement la plus grave que j'ai constatée (et je ne suis pas le seul) vient du moteur JavaScript : sous IE6 et IE7, le fait d'oublier d'enlever la dernière virgule (',') après une chaîne JSON a toujours planté le script

```
var oData : {
    value1:"blah",
    value2:"blah", // === boom
};
```

Sous IE8 comme sur tous les autres navigateurs, cette syntaxe est valide. Or, si vous vérifiez avec le mode de compatibilité IE7 sous IE8, **vous ne verrez pas cette erreur** qui casse votre JavaScript et qui enlève donc des fonctionnalités majeures de votre site, simplement parce que vous n'êtes pas dans les bonnes conditions pour tester.

3. Les installations multiples d'Internet Explorer



Internet Explorer est à tel point implanté dans son OS qu'il est en théorie impossible d'en installer plusieurs versions (ce qui leur avait même valu des problèmes avec la justice : [Lien 63](#)). Mais plusieurs softs ont pallié cette situation et permettent **d'exécuter plusieurs versions d'IE** : par exemple IETester ([Lien 64](#)), IECollection ([Lien 65](#)) ou Multiple IE ([Lien 66](#)). Le problème, c'est que ces différentes instances d'IE continuent de partager certaines DLL et autres appels système. Ça n'a l'air de rien, mais concrètement pour le développeur Web ça l'empêche de voir certaines choses.

3.1. Plugins

Les interactions avec des plugins populaires comme *Flash* ou *Java* sont problématiques :

- j'ai déjà vu *Flash* planter sans que IE dans un *IETester* ne s'en aperçoive, le fallback que nous implémentions en *Flash* (la sélection multiple de fichiers pour un uploader) était essentiel, mais le développement était pénible : on s'est rendu compte au bout d'un moment que *Flash* ne répondait plus ou mal et qu'il fallait tuer le processus *Flash* pour que cela remarche. Avec un

IE natif, pas de problème : lorsque *Flash* plante, il se relance correctement, ce qui accélérât le développement ;

- l'interaction avec une applet *java* (toujours pour un uploader, en drop de fichiers cette fois) était tout simplement impossible à tester avec plusieurs IE dans le même *IETester*. Une machine virtuelle, c'est déjà lourd à lancer, alors imaginez trois applets en parallèle qui essaient de s'afficher sur la même zone de l'écran. Les fichiers n'étaient pas réceptionnés dans la bonne instance d'IE.

3.2. Fontes et couleurs

Pour le rendu des fontes il vous faudra typiquement vérifier votre site sous un Safari sur Mac et sur un IE6 sous XP : le **Safari Mac rend les polices exactement comme votre graphiste l'a prévu** sur sa belle maquette *Photoshop*. C'est la vision qu'il a du site et c'est sûrement celle que les décideurs ont également...

Le couple terrible, IE6-XP qui est encore là pour longtemps, est lui à *l'extrémité opposée du spectre des rendus* et correspond au pire de ce que verront vos visiteurs. Les versions supérieures de IE et des navigateurs ne font qu'améliorer le rendu sans jamais totalement atteindre la perfection du couple Safari / OSX.



Au moins quatre rendus par police

J'ajoute que selon la taille de police, l'activation ou pas de l'option lissage, le type de police et les caractères utilisés, certains mots peuvent devenir illisibles ou du moins pénibles à lire sur les Windows. Lorsque vous décidez de l'utilisation d'une police, il vous faudra d'abord **vérifier son rendu final sur au moins XP, Vista ou Seven et enfin Mac OS**. Soit vous maintenez trois machines physiques avec un OS sur chacune, soit vous utilisez les machines virtuelles (ou plus pénible : des machines en dual boot).

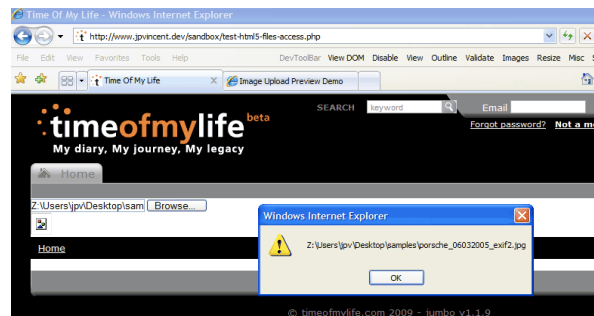
Le rendu des couleurs : j'avoue ne pas très bien savoir comment faire ici. Mac et les différents Windows rendent les couleurs différemment, discutez-en deux minutes avec un graphiste, ça lui laissera le loisir de vous montrer qu'il en sait plus que vous. Or avec les machines virtuelles, on affiche un Windows sur un Mac, je n'ai aucune idée de ce que cela donne réellement. Pour en rajouter un peu, je n'obtiens pas les mêmes couleurs selon que je sois sur l'écran du MacBook ou sur le second écran... Devant mon incapacité à vous expliquer comment tout cela fonctionne, je ne peux que citer une des bonnes pratiques généralement accolées à l'accessibilité : **inutile de jouer avec des nuances de couleurs, préférez de forts contrastes**.

3.3. Être à jour

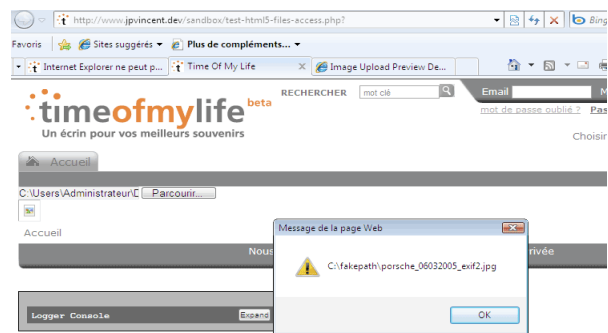
Microsoft patche régulièrement ses OS ainsi que ses IE. Rarement mais sûrement certaines choses peuvent **s'arrêter de fonctionner suite à ces patches**. Exemple concret : dans les IE, on pouvait récupérer le chemin local d'un fichier sélectionné par un input type=file. Ceci permettait d'afficher une miniature d'une image avant upload, fonctionnalité d'ailleurs à nouveau permise par

HTML5. En 2009, un patch est passé et a fait disparaître cette possibilité, après dix ans de bons et loyaux services (voir les screenshots que j'avais faits : [Lien 67](#)).

- **Avant le patch :**



- **Après le patch :**



Les packs d'IE ne permettent pas de rester à jour, ou alors vous dépendez de leur bon vouloir. Avoir une machine virtuelle permet d'utiliser le mécanisme d'auto-update de Microsoft.

4. Dégradation gracieuse et fallbacks

4.1. Les anciens et les futurs navigateurs

Outre Internet Explorer, certaines versions de Firefox valent encore la peine d'être testées avec plusieurs versions. Ça a été le cas pendant longtemps pour FF2, c'est encore peut-être le cas sur vos sites pour FF3.0 et à l'heure où j'écris, c'est certainement vrai pour FF3.5, malgré le fait que l'on soit en 3.6 depuis longtemps.

Les *bugfixs* passés dans ces releases impactent rarement le travail des développeurs (mais ça arrive), mais il y a surtout eu des ajouts de fonctionnalités majeures (rangées sous l'ombrelle HTML5) ou l'ajout de supports CSS3 : il faut que vous testiez au moins une fois pendant le développement comment se comporte votre fallback ou votre *graceful degradation* sur ces anciennes versions.

Il n'est peut-être pas utile de conserver des versions de Chrome et Opéra, l'un car il force l'auto-update, l'autre parce qu'il est rare, mais c'est à vous de juger.

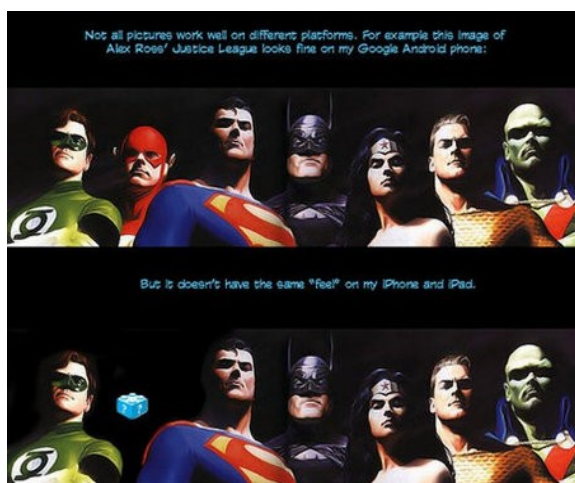


En revanche, il est vital d'avoir **les dernières versions d'IE9 et FF4** pour tester régulièrement votre site avec : faites corriger les bogues par Microsoft ou par Mozilla avant que vos utilisateurs ne tombent dessus et vous fassent changer votre code !

Il n'est peut-être pas nécessaire d'avoir une machine virtuelle dédiée à une version passée et future des navigateurs hors IE, car il me semble que différentes versions de FF savent cohabiter et que les bêta/preview d'IE savent également s'installer sans conséquence. Mais il vaut peut-être mieux **jouer la prudence** là-dessus : ce sont des versions bêta et avec les nouvelles fonctionnalités type accélération matérielle, les navigateurs s'intègrent de plus en plus au système, il me semble donc évident que **toute nouvelle installation met en péril l'intégrité des autres versions du même navigateur**. Une fois que le mal est fait, difficile de revenir en arrière, tandis que sur une machine virtuelle, vous pouvez isoler le problème et même **revenir en arrière grâce aux fonctionnalités de snapshot** à exécuter avant toute installation hasardeuse.

4.2. Plugins et add-ons

Votre site dépend parfois de *Flash* ou *Java* ? Il affiche des publicités ?



Il vous faut au moins une machine où *Flash* et *Java* ne sont pas installés, afin de **tester de manière réaliste votre contenu alternatif ou le fallback de votre fonctionnalité**. On ne développe pas ce qu'on ne voit pas.

Concernant *Flash* et c'est valable avec *Java* : il existe des différences entre Flash 9 (encore répandu) et Flash 10.x. Les flasheurs développent et testent toujours avec la dernière version : ayez donc une machine avec une ancienne installation de *Flash* pour **vérifier rapidement que le *Flash* qu'on vous livre marche pour tout le monde**.

Votre Firefox est peut-être comme le mien : c'est devenu un navigateur de développement sur lequel sont installés une dizaine de plugins dont le gourmand Firebug. Après quelques heures de développement, mon Firefox est à la limite du praticable sur certaines pages chargées. Un tour régulier sur un Firefox sans aucun plugin dans une machine virtuelle me permet **de vérifier (et non pas de supposer) si c'est un plugin ou mon code qui fait ramer**

le navigateur.

5. Quelles contraintes ?

Vous êtes probablement d'accord avec le principe d'être dans de bonnes conditions pour tester, mais vous avez peur que cela demande **du budget et de la ressource machine** ? C'est le cas, mais rien d'insurmontable. Concernant votre machine de développement d'abord.

- **Mac OS** : il faudra le tester de toute façon, ne serait-ce que pour vérifier ce que voient **les graphistes et les journalistes...** Pendant longtemps il n'y a pas eu de machine virtuelle capable de le faire tourner, mais il semblerait que cela soit aujourd'hui possible ([Lien 68](#)), donc si vous voulez rester sous Windows ou Linux, c'est peut-être possible, à vos risques et périls. Pour ma part et depuis quatre ans, j'ai convaincu mes employeurs de payer 50 % plus cher mon matériel pour être dans des conditions de test réalistes. Pour les managers qui doivent justifier ce budget : outre la qualité, un développeur sous Mac est un développeur qui restera plus longtemps chez vous.
- **4 Go de RAM** : pendant le développement, il faut tourner avec au moins un IE6 (ou 7) sous XP en permanence ouvert en plus de l'environnement de développement normal (Firefox, éditeurs, divers logiciels...). **Avec 2 Go, c'est jouable mais pénible**, du coup certains développeurs ferment à jamais la machine. 4 Go, c'est mieux et ça permet de supporter ces jours pénibles, généralement en début et en fin de projet, où vous allez lancer successivement toutes vos machines virtuelles pour valider des choix techniques ou le développement lui-même.
- **Un disque rapide** : deux OS qui cohabitent, ça fait énormément d'appels au disque dur, surtout que vous allouerez peu de RAM à Windows (512 Mo maximum en général), donc ce dernier utilisera la mémoire virtuelle (le disque donc). Si vous n'avez que 2 Go de RAM et un disque de portable à 5400 tr/min, la situation sera intenable. Essayez d'upgrader à 7200 (la différence est flagrante), voire de le remplacer par un SSD, technologie qui commence à être abordable. Je n'ai pas essayé le dernier MacBook AIR, mais la mémoire Flash qu'il utilise en remplacement semble parfaite dans ce cas ([Lien 69](#)).

Concernant le software.

- **XP et Seven** : il vous faut au moins ces deux OS pour installer de IE6 à IE9. Concernant les licences, chez Yahoo! nous utilisons des licences dites corporate (valables pour des milliers de postes), donc pas de problème. En PME, la pratique généralement admise est d'acheter la licence, d'installer... puis de cloner l'image de la machine autant de fois qu'il y a d'environnements et de développeurs. Honnêtement, **je ne sais pas si c'est prévu par la licence**. Si vous émulez à partir de Seven avec Virtual PC, Microsoft vous offre des images d'installation (limitées dans le temps) ([Lien 70](#)) parfaites pour les développeurs Web.

- **Une machine virtuelle Mac. Payant** : j'utilise Parallels Desktop ([Lien 71](#)) qui me satisfait pleinement mais qui a la réputation d'être gourmand en ressources. J'ai entendu du bien de VMWare ([Lien 72](#)). Dans tous les cas comptez **moins de 100 euros**. **Gratuit** : VirtualBox ([Lien 73](#)) est le leader de l'open source, mais n'était pas sorti lorsque j'ai commencé à utiliser Parallels il y a quatre ans, je ne peux donc pas en parler. **Si vous êtes sur Windows Seven, vous pouvez utiliser Virtual PC ([Lien 74](#)).**

Les machines virtuelles vous fournissent généralement des images préinstallées. Mais elles ne sont jamais en français et bien sûr ne dispensent pas de fournir une clé Windows valide. Veillez aussi à ce que les images de XP fournies contiennent bien IE6 : on ne peut pas désinstaller IE7 pour installer IE6.

6. La matrice de tests

Je vous conseille de suivre la matrice de Yahoo! ([Lien 75](#)), mise à jour deux à trois fois par an. Voici en tout cas le contenu de mes machines après plusieurs années de pratique :



Vista et FF2-3 ne sont peut-être plus utiles à posséder, mais sinon toutes ces combinaisons tiendraient en cinq machines. Cela donne accès à :

- pas de Flash/java, Flash 9 et un Flash à jour partout ailleurs ;
- IE 6, 7 et 8 sur XP, IE8 sur Seven ;

- FF3.0 et FF 3.5 (qui n'apparaît pas) ;
- IE9 bêta à jour, FF4 bêta à jour. Un FF *nightly build* pour vérifier des bugfix à l'occasion ;
- ces machines comportent presque toutes un Chrome et un Opéra à jour, en plus du Chrome / Opéra / Safari de mon Mac. Oui il y a parfois des différences, pas uniquement visuelles, entre une version Mac et Windows d'un même navigateur.

Cela fait beaucoup de combinaisons et il en manque encore : par exemple si je dois un jour tester le site sur IE6 sans Flash, je n'ai pas de machine toute prête. Qu'à cela ne tienne, il me suffit de cloner ou de faire un snapshot de ma machine IE6 puis de désinstaller Flash pour vérifier mon bogue.

Il existe aussi des équipes Web qui maintiennent des PC avec toutes ces combinaisons, accessibles physiquement ou depuis un VNC ([Lien 76](#)). De ce que j'en ai vu, c'est une solution peu pratique que les développeurs finissent par ne plus utiliser, préférant coder à l'aveugle plutôt que de se déplacer ou subir des lags réseau... Oui les développeurs sont comme ça, d'où l'intérêt d'investir dans du matériel !

7. Conclusion

- Si vous vous considérez comme un professionnel du Web, vous devez d'investir dans un environnement de test fiable.
- Il y a de vraies différences de comportement et de rendu entre un IE natif, des packs d'IE et entre OS.
- Il y a un grand nombre de combinaisons possibles (quatre OS, cinq navigateurs, plusieurs sous-versions, Flash...) et les différences iront en s'accroissant (nouveaux navigateurs, adaptation à chaque OS...).
- La seule solution fiable et souple est d'avoir plusieurs machines virtuelles.

Retrouvez l'article de Jean-Pierre Vincent en ligne : [Lien 77](#)

Ajax et le référencement

Par nature, une requête AJAX fait appel à JavaScript pour obtenir une ressource. Donc les moteurs de recherche ne sont pas capables aujourd'hui de suivre ces liens.

1. Exemple de travail

Prenons l'exemple d'un site Web de trois pages utilisant AJAX pour les charger sans recharger la totalité de la page affichée (habillage et navigation sont conservés).

La navigation pourrait ressembler à ceci :

```
<ul id='nav'>
<li>
<a href="javascript:load('page1.html') ">Page1</a>
</li>
<li>
<a href="javascript:load('page2.html') ">Page2</a>
</li>
<li>
<a href="javascript:load('page3.html') ">Page3</a>
</li>
</ul>
```

Dans un site Web traditionnel, la navigation ressemblerait plutôt à ceci :

```
<ul id='nav'>
<li><a href="page1.html">Page 1</a></li>
<li><a href="page2.html">Page 2</a></li>
<li><a href="page3.html">Page 3</a></li>
</ul>
```

Pour pallier le problème du référencement, les moteurs de recherche devraient avoir accès à cette deuxième version.

Les deux ne sont pas incompatibles techniquement. Il suffit de produire un code (X)HTML pour les moteurs de recherche et de court-circuiter les liens avec JavaScript pour faire des chargements via AJAX.

```
function initNav() {
    var nav = document.getElementById('nav');
    var liens = nav.getElementsByTagName('A');
    for(var i=0;i<liens.length;i++){
        liens[i].onclick = load;
    }
}
```

La fonction initNav doit être exécutée au chargement de la page, la fonction load doit interrompre le traitement par défaut du clic et nous avons résolu notre premier problème.

2. Fonctionnement sans JavaScript

Si JavaScript est désactivé, les URL définies dans le href vont être utilisées quand on clique sur les liens. Le navigateur devra recevoir un contenu de page complet (de

même pour les moteurs de recherche).

Dans le cadre d'une application AJAX, les contenus que l'on souhaite charger sont partiels (pas de DOCTYPE, ni de body), juste le contenu, c'est-à-dire des fragments (X)HTML.

Pour pallier ce problème il faut pouvoir produire les deux types de contenu depuis le serveur. Le plus simple étant de séparer les fragments (X)HTML dans des fichiers distincts et de travailler avec un modèle (X)HTML contenant l'habillage du site et sa navigation mais sans contenu.

Exemple de structure du site :

```
model.html
fragments/page1.html
fragments/page2.html
fragments/page3.html
index.php
```

La page model.html pourrait ressembler à ceci :

```
<!DOCTYPE ...>
<html ...>
<head>...</head>
<body>
<ul id="nav">
<li><a href="page1.html">Page 1</a></li>
<li><a href="page2.html">Page 2</a></li>
<li><a href="page3.html">Page 3</a></li>
</ul>
<div id="contenu">[[CONTENU]]</div>
</body>
</html>
```

On peut utiliser PHP pour produire les pages avec un code du genre :

```
$page = $_GET['page'];
$fragment = file_get_contents("fragments/$page");
$model = file_get_contents('model.html');
echo str_replace('[[CONTENU]]', $fragment, $model);
```

Les URL produisant des pages complètes sont de la forme : index.php?page=page1.html, celles permettant d'obtenir des fragments sont ainsi : fragments/page1.html (celles-ci ne seront jamais indexées).

En utilisant l'URL-rewriting ([Lien 78](#)), on peut facilement transformer les URL comme page1.html en index.php?page=page1.html.

Ainsi, nous avons une application totalement fonctionnelle sans JavaScript et qui peut être référencée par les moteurs de recherche.

3. Gestion de l'URL avec AJAX

Pour éviter le rechargement de la page avec AJAX, on ne peut pas se permettre d'en changer l'URL, ce qui rentre en conflit avec ce qui a été dit plus haut et selon la règle qu'à chaque contenu doit correspondre une URL et réciproquement.

Heureusement, il est possible de modifier l'ancree d'une URL sans recharger la page (les ancres servant initialement à se déplacer dans une même page).

On peut modifier l'URL de la barre d'adresse du navigateur en JavaScript sans recharger la page en appliquant une ancre avec `document.location.replace('#page1.html')`. Ceci doit être fait après chaque requête AJAX qui charge un nouveau contenu.

Les URL de notre application AJAX ressemblent donc à :

```
/#page1.html  
/#page2.html  
/#page3.html
```

Il est important que l'application AJAX examine l'URL du

navigateur lors du premier chargement de la page de manière à afficher le bon contenu. Ces URL comportant des ancres peuvent être utilisées suite à la mise en favoris d'une page par exemple.

4. Conflit lors de l'arrivée sur le site par une page référencée

En accédant à un site par l'URL `page2.html`, référencée par un moteur de recherche, étant donné l'absence d'ancre, l'application AJAX pourrait traiter ce cas et afficher le contenu de la page2. Cependant, lorsque l'on navigue dans le site sur la page3, on aurait une URL `page2.html#page3.html`, ce qui n'est pas très cohérent d'une part et qui contredit la règle d'unicité entre le contenu et les URL d'autre part.

L'application AJAX devrait donc faire une redirection en JavaScript vers l'URL `/#page2.html` au plus tôt, avant l'affichage du corps de la page (en plaçant un script dans l'entête).

5. Exemple fonctionnel

Voici un petit exemple fonctionnel à placer sur votre serveur (qui doit supporter le .htaccess pour l'URL-rewriting) : [Lien 79](#) (2.58 ko).

Retrouvez l'article de Marc Chappuis en ligne : [Lien 80](#)

Comment centrer ses widgets "fenêtre" ?

On peut constater qu'à chaque fois qu'on crée une fenêtre Tk, celle-ci est positionnée un peu aléatoirement à l'écran. Utilisez la procédure **centrer_widget** du programme ci-dessous pour centrer vos widgets.

Centrer une fenêtre Tk

```
#!/usr/bin/perl
use warnings;
use strict;
use Tk;      # Appel du module Tk

# Programme principal
# Création de la fenêtre
my $fenetre = new MainWindow(
    -title      => 'Première fenêtre Tk',
    -background => 'white',
);

# Taille minimale de ma fenêtre
$fenetre->minsize( 300, 100 );

my $message_accueil = "Bonjour tout le
monde\n\nWelcome dans le monde magnifique de
Perl/Tk\n\n";

# Affichage d'un texte
my $label_accueil = $fenetre->Label(
    -text      => $message_accueil,
    -background => 'white',
)->pack();

# Affichage d'un bouton pour fermer la fenêtre
my $bouton = $fenetre->Button(
    -text      => 'Détruire la fenêtre',
    -command => sub { exit; },
)->pack();

# Centrer ma fenêtre
centrer_widget($fenetre);

MainLoop();      # Obligatoire

#####
# But : Centrer un widget automatiquement
#####
sub centrer_widget {
    my ( $widget ) = @_;

    # Height and width of the screen
    my $largeur_ecran = $widget->screenwidth();
    my $hauteur_ecran = $widget->screenheight();

    # update le widget pour récupérer les vraies
    dimensions
    $widget->update;
    my $largeur_widget = $widget->width;
    my $hauteur_widget = $widget->height;
```

```
# On centre le widget en fonction de la taille
de l'écran
my $nouvelle_largeur = int( ( $largeur_ecran -
$largeur_widget ) / 2 );
my $nouvelle_hauteur = int( ( $hauteur_ecran -
$hauteur_widget ) / 2 );
$widget->geometry($largeur_widget . "x" .
$hauteur_widget
    . "+$nouvelle_largeur+$nouvelle_hauteur");

$widget->update;

return;
}
```

Comment créer des onglets ?

Pour créer des onglets, il faut utiliser le widget **NoteBook**. Pour cela, il est nécessaire d'appeler le module **Tk::NoteBook** dans votre script. Il est installé avec le module Tk.

Vous devez dans un premier temps créer un bloc-notes. C'est le cadre sur lequel doivent être positionnés les onglets.

```
my $blocnote = $fenetre_parent->NoteBook();
```

Ensuite, il suffit de créer les onglets nécessaires.

```
my $onglet1 = $blocnote->add('NomOnglet', -label
=> 'onglet 1');
```

Chaque onglet doit avoir un nom qui lui sera unique. Maintenant, sur chaque onglet, vous pouvez faire ce que vous voulez, comme créer un cadre ou un bouton.

Voici un exemple :

```
#!/usr/bin/perl
use strict;
use warnings;
use Tk;
use Tk::NoteBook;

my $fenetre_principale = MainWindow->new( -title
=> 'onglet', -background => 'yellow', );
$fenetre_principale->minsize ( 300, 300);
$fenetre_principale->Label ( -text => "exemple
d'onglets",)->pack();

# blocnote
my $blocnote = $fenetre_principale->NoteBook(
    -backpagecolor => 'white',
    -inactivebackground => 'pink',
)->pack( qw/-fill both -expand 1/ );

# onglets
my $onglet1 = $blocnote->add('onglet1', -label =>
'onglet 1,');
my $onglet2 = $blocnote->add('onglet2', -label =>
```

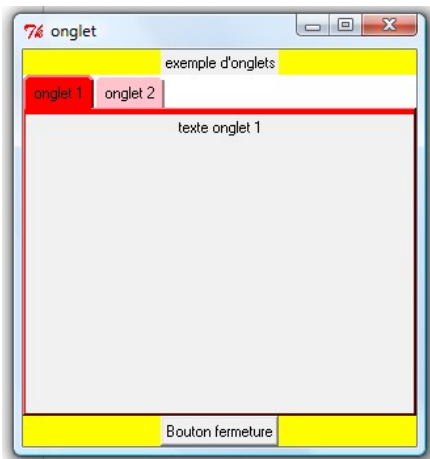


```
'onglet 2',);

$onglet1->Label( -text => 'texte onglet 1',)->pack();
$onglet2->Label( -text => 'texte onglet 2',)->pack();

$fenetre_principale->Button (
    -text => 'Bouton fermeture',
    -command => sub { exit; },
)->pack();

MainLoop;
```



Comment modifier la couleur de fond d'un onglet ?

Vous avez sans doute remarqué qu'il est impossible de mettre une couleur de fond avec l'option **-background** lorsque l'on crée un onglet avec la méthode **add**. Prenons par exemple ce code.

Voici un exemple :

```
#!/usr/bin/perl
use strict;
use warnings;
use Tk;
use Tk::NoteBook;

my $fenetre_principale = MainWindow->new( -title
=> 'onglet', -background => 'yellow', );
$fenetre_principale->minsize ( 300, 300);
$fenetre_principale->Label ( -text => "exemple
d'onglets",)->pack();

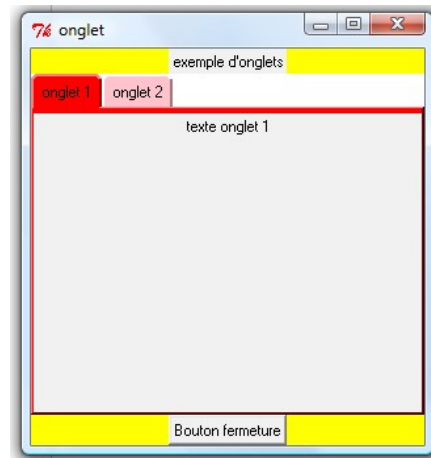
# blocnote
my $blocnote = $fenetre_principale->NoteBook(
    -background => 'red',
    -backpagecolor => 'white',
    -inactivebackground => 'pink',
)->pack( qw/-fill both -expand 1/ );

# onglets
my $onglet1 = $blocnote->add('onglet1', -label =>
'onglet 1', );
my $onglet2 = $blocnote->add('onglet2', -label =>
'onglet 2',);

$onglet1->Label( -text => 'texte onglet 1',)->pack();
$onglet2->Label( -text => 'texte onglet 2',)->pack();
```

```
$fenetre_principale->Button (
    -text => 'Bouton fermeture',
    -command => sub { exit; },
)->pack();

MainLoop;
```



Si vous essayez de mettre l'onglet 1 en vert ainsi

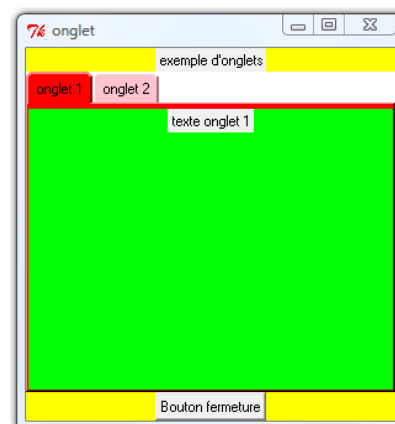
```
my $onglet1 = $blocnote->add('onglet1', -label =>
'onglet 1', -background => 'green');
```

vous obtiendrez le message d'erreur suivant : *Bad option '-background' at C:/Perl/site/lib/Tk.pm line 250..* Normal, cette option n'est pas disponible ! Lisez la documentation de **Tk::NoteBook** !

Néanmoins, il y a une parade toute simple. La méthode **add** retourne un objet de type **Frame**. Il suffit de le récupérer et d'y appliquer les options dont on a besoin (option de **Tk::Frame**).

```
my $onglet1 = $blocnote->add('onglet1', -label =>
'onglet 1',);
$onglet1->configure( -background => 'green',);
```

Voici maintenant ce que l'on obtient.



Génial non ?!!

Qu'est-ce qu'une boîte de dialogue ?

Il peut être utile d'avoir une interactivité avec l'utilisateur de votre application via un pop-up qui aura pour but de signifier une erreur, d'émettre un warning (une attention particulière), de demander une confirmation, etc. Pour ce faire, le module Tk dispose de quatre boîtes de dialogue standard faciles à utiliser. Chacune d'elles sera illustrée dans les questions ci-dessous. Notez qu'il vous est également possible de créer vos propres pop-up via la méthode Toplevel et de l'habiller à votre guise, voire même de créer un widget composite.

Comment créer une boîte de dialogue avec messagebox ?

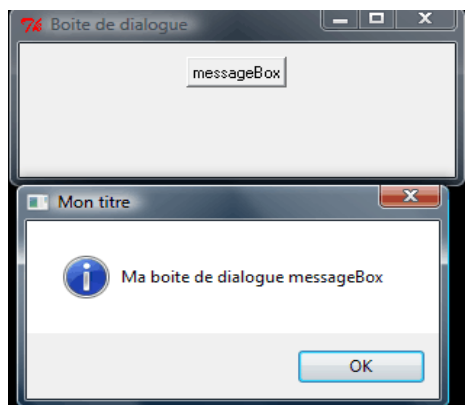
Pour créer une boîte de dialogue **messageBox**, il faut utiliser la méthode **messageBox** interne à Tk. **messageBox** n'est pas un widget composite comme les autres boîtes de dialogue : vous n'aurez pas besoin de faire appel à un module particulier. Voici un exemple illustrant son utilisation :

```
#!/usr/bin/perl
use warnings;
use strict;
use Tk;

my $mw = new MainWindow( -title => "Boite de
dialogue", );
$mw->minsize( 300,100 );

my $reponse = $mw->Button(
    -text    => 'messageBox',
    -command => sub {
        my $reponse_messageBox = $mw->messageBox(
            -icon    => 'info',
            -title   => 'Mon titre',
            -type    => 'OK',
            -message => 'Ma boite de dialogue
messageBox',
        );
    }
)->pack( qw/ -padx 10 -pady 10 / );

MainLoop();
```



L'aspect de la fenêtre est dépendant de votre système d'exploitation. A ce jour, il est impossible de modifier la couleur de fond de la fenêtre, du bouton, de changer

l'icône du pop-up, ce qui peut être un inconvénient.

\$reponse contient la réponse de l'utilisateur. Dans notre cas, elle sera égale à **OK** si l'utilisateur clique sur le bouton OK. Elle aurait pu être égale à **CANCEL** si l'on avait mis **-type** à **'OKCANCEL'** et cliqué sur le bouton **annuler**.

- L'option **-icon** permet de spécifier le type d'image qui sera affichée à la gauche du message.
- L'option **-type** permet de spécifier le type de bouton. Les choix sont : 'AbortRetryIgnore', 'Ok', 'OkCancel', 'RetryCancel', 'YesNo' or 'YesNoCancel'

Pour plus d'informations, regardez la documentation officielle.

Comment créer une boîte de dialogue avec MsgBox ?

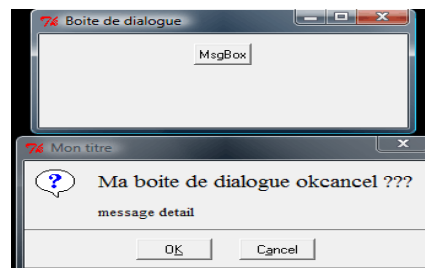
Pour créer une boîte de dialogue **MsgBox**, il faut utiliser le module **Tk::MsgBox**. Inutile de chercher à l'installer si Tk l'est car il fait partie du core de Tk. Voici un exemple illustrant son utilisation :

```
#!/usr/bin/perl
use warnings;
use strict;
use Tk;
use Tk::MsgBox;

my $mw = new MainWindow( -title => "Boite de
dialogue", );
$mw->minsize( 300,100 );

$mw->Button(
    -text    => 'MsgBox',
    -command => sub {
        my $msgbox = $mw->MsgBox(
            -title   => 'Mon titre',
            -type    => 'okcancel',
            -message => 'Ma boite de dialogue
okcancel',
        );
        my $reponse = $msgbox->Show;
        print "Reponse : $reponse\n";
    }
)->pack( qw/ -padx 10 -pady 10 / );

MainLoop();
```



L'aspect de la fenêtre est dépendant de votre système d'exploitation. A ce jour, il est impossible de modifier la couleur de fond de la fenêtre, du bouton, de changer l'icône, ce qui peut être un inconvénient.

\$reponse contient la réponse de l'utilisateur. Dans notre cas, elle sera égale à **OK** ou à **CANCEL** en fonction du clic de l'utilisateur.

- L'option **-icon** permet de spécifier le type d'image qui sera affichée à la gauche du message.
- L'option **-type** permet de spécifier le type de bouton. Les choix sont : **abortretryignore**, **ok**, **okcancel**, **retrycancel**, **yesno** ou **yesnocancel**.
- L'option **-detail** permet de spécifier un message supplémentaire.

Pour plus d'informations, regardez la documentation officielle.

Comment créer une boîte de dialogue avec DialogBox ?

Pour créer une boîte de dialogue **DialogBox**, il faut utiliser le module **Tk::DialogBox**. Inutile de chercher à l'installer si Tk l'est car il fait partie du core de Tk. C'est un widget composite (module Tk issu d'un autre widget). On peut dans cette boîte de dialogue rajouter d'autres widgets, ce qui nous laisse une plus grande marge de manœuvre pour le design de notre boîte de dialogue. Nous pouvons également mettre le texte que l'on veut dans nos boutons et surtout les configurer pour changer leur couleur via la méthode **Subwidget** par exemple.

Voici un exemple illustrant son utilisation.

```
#!/usr/bin/perl
use warnings;
use strict;
use Tk;
use Tk::DialogBox;

my $mw = new MainWindow( -title => "Boite de
dialogue", );
$mw->minsize( 300,100 );

$mw->Button(
    -text      => 'DialogBox',
    -command => [\&BoiteDialogBox, $mw],
)->pack( qw/ -padx 10 -pady 10 / );

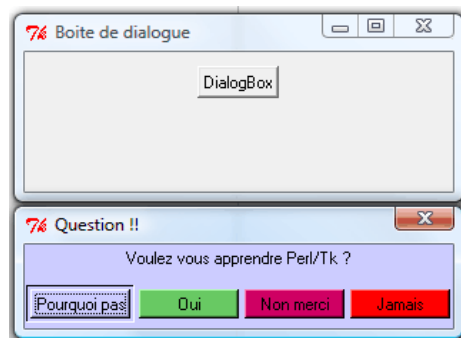
MainLoop();

sub BoiteDialogBox {
    my $widget = shift;
    my $dialog_box = $widget->DialogBox(
        -title      => 'Question !!',
        -buttons    => ["Pourquoi pas", "Oui",
"Non merci", "Jamais"],
        -background => '#CCCCFF',
        -default_button => 'Oui',
    );

    # Rajout d'un label
    $dialog_box->Label(
        -text      => "Voulez-vous apprendre
Perl/Tk ?",
        -background => '#CCCCFF',
    )->pack();

    # Configuration des boutons
    $dialog_box->Subwidget("B_Oui")->configure(
        -background => '#68C963', );
}
```

```
$dialog_box->Subwidget("B_Non merci")->
configure( -background => '#CF0063', );
$dialog_box->Subwidget("B_Jamais")->
configure( -background => 'red', );
my $reponse = $dialog_box->Show();
print "Reponse : $reponse\n";
}
```



Comme vous pouvez le constater, il est possible de configurer notre boîte de dialogue et de l'habiller plus sympathiquement par rapport aux autres boîtes de dialogue (**MsgBox** et **messageBox**).

\$reponse contient la réponse de l'utilisateur. Dans notre cas, elle sera égale à **Pourquoi pas**, **Oui**, **Non merci** ou à **Jamais** en fonction du clic de l'utilisateur.

Notez l'usage de **Subwidget** pour obtenir la référence du widget Button voulu. Dans la documentation officielle, il est stipulé d'écrire `$dialog_box->Subwidget("B_TEXTEBOUTON")` pour obtenir la référence du bouton afin de la configurer à souhait.

Pour plus d'informations, regardez la documentation officielle.

Comment créer une boîte de dialogue avec Dialog ?

Pour créer une boîte de dialogue **Dialog**, il faut utiliser le module **Tk::Dialog**. Inutile de chercher à l'installer si Tk l'est car il fait partie du core de Tk. C'est un widget composite (module Tk issu d'un autre widget **Tk::DialogBox**). Tout ce qui a été expliqué pour **Tk::DialogBox** est applicable à ce module.

Voici un exemple illustrant son utilisation.

```
#!/usr/bin/perl
use warnings;
use strict;
use Tk;
use Tk::Dialog;

my $mw = new MainWindow( -title => "Boite de
dialogue", );
$mw->minsize( 300,100 );

$mw->Button(
    -text      => 'Dialog',
    -command => [\&BoiteDialogMsgBox, $mw],
)->pack( qw/ -padx 10 -pady 10 / );

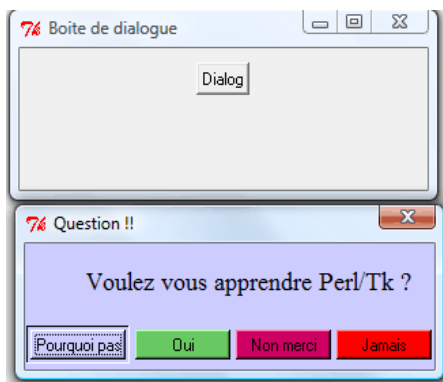
MainLoop();
```

```

sub BoiteDialogMsgBox {
    my $widget = shift;
    my $dialog = $widget->Dialog(
        -title      => 'Question !!',
        -buttons    => ["Pourquoi pas", "Oui",
            "Non merci", "Jamais"],
        -background => '#CCCCFF',
        -text       => "Voulez-vous apprendre
Perl/Tk ?",
    );

    # Configuration des boutons
    $dialog->Subwidget("B_Oui")->configure(
        -background => '#68C963', );
    $dialog->Subwidget("B_Non merci")->
configure( -background => '#CF0063', );
    $dialog->Subwidget("B_Jamais")->configure(
        -background => 'red', );
    my $reponse = $dialog->Show();
    print "Reponse : $reponse\n";
}

```



Comme vous pouvez le constater, il est possible de configurer notre boîte de dialogue et de l'habiller plus sympathiquement comparé aux autres boîtes de dialogue (**MsgBox** et **messageBox**).

\$reponse contient la réponse de l'utilisateur. Dans notre cas, elle sera égale à **Pourquoi pas**, **Oui**, **Non merci** ou à **Jamais** en fonction du clic de l'utilisateur.

Notez l'usage de **Subwidget** pour obtenir la référence du widget Button voulu. Dans la documentation officielle, il est stipulé d'écrire **\$DialogBox->Subwidget("B_TEXTEBOUTON")** pour obtenir la référence du bouton afin de la configurer à souhait.

A la différence de **Tk::DialogBox**, il y a une option **-text** permettant de spécifier son texte.

Pour plus d'informations, regardez la documentation officielle.

Comment changer l'habillage de mon canevas ?

Pour changer la couleur de son canevas, il est possible de modifier les couleurs de ce dernier via les options prédéfinies du module **Tk::Canvas**. Sachez qu'il est également possible d'appliquer une couleur dégradée grâce au module **Tk::Canvas::GradientColor**.

Le sujet est traité à la question : Comment créer un canevas avec une couleur de fond dégradée ? ([Lien 81](#))

Comment changer l'habillage de mon bouton ?

Pour changer la couleur de son bouton, il est possible de modifier les couleurs de ce dernier via les options prédéfinies du module **Tk::Button**. Sachez qu'il est également possible d'appliquer une couleur dégradée grâce au module **Tk::ColoredButton**.

Le sujet est traité à la question : Comment créer un bouton avec une couleur de fond dégradée ? ([Lien 82](#))

Comment configurer des couleurs avec setPalette, bisque ?

La réponse ci-dessus est issue du livre Introduction à Perl/Tk ([Lien 83](#)) qui n'est malheureusement plus disponible en français.

- Configuration des couleurs

La méthode **SetPalette** permet à votre application d'utiliser automatiquement des couleurs en fonction d'une couleur donnée.

```
$widget->setPalette('couleur');
```

La couleur de fond de **\$widget** sera de la couleur spécifiée et les couleurs de tous les autres widgets seront calculées à partir de celle-ci. Ainsi, si le contour d'un bouton est plus clair que le fond, il apparaîtra dans une teinte plus claire que la couleur choisie. Cette méthode affecte l'application entière, même si on l'a appelée sur un widget plutôt que sur la fenêtre.

Les couleurs de certaines options sont configurables ; il suffit de fournir le nom de l'option, puis la couleur que l'on souhaite lui associer. Le fragment de code suivant, par exemple, met en rouge l'avant-plan de l'application et en bleu le fond :

```
$b->setPalette("background" => "blue",
"foreground" => "red");
```

- Configuration de couleur prédéfinie

La méthode bisque utilise la configuration de couleur "bisque" pour toute l'application. L'appel **\$widget->bisque()** est équivalent à **\$widget->setPalette("bisque")**

Comment appliquer un thème (un habillage) à mes widgets ?

Il n'existe pas de thème en Perl Tk permettant de changer l'aspect de l'interface graphique. Dans une application graphique, le plus important n'est pas le rendu et sa beauté, mais d'abord le résultat que l'on attend, donc son but. Néanmoins, il est toujours agréable de changer l'habillage de ses fenêtres, de ses widgets !

L'aspect d'une interface graphique dépend d'abord du système d'exploitation sous lequel tourne l'application. Le

rendu sera différent que l'on soit sous Windows 2000, XP, Vista, Seven, Linux, Mac, BSD, etc. Mais il est tout de même possible de changer les couleurs de nos widgets, les épaisseurs des bordures...

Pour changer l'aspect de nos widgets, il faut modifier soit- même à chaque création d'un widget les options *-background*, *-foreground*, *-disabledbackground*, *-highlightbackground*... De plus, il faut vérifier dans la documentation de chaque widget la disponibilité des options. Ce travail peut être fastidieux lorsque notre application contient plusieurs widgets.

Pour effectuer un changement **global** à notre application, nous pouvons utiliser la méthode `SetPalette` ([Lien 84](#)). Cette méthode est efficace, mais le rendu obtenu n'est pas toujours celui souhaité. De plus, il est impossible de le personnaliser. Une autre technique consiste à utiliser le module standard de Tk `Tk::CmdLine` ([Lien 85](#)). Nous n'allons pas expliquer son fonctionnement qui n'est pas toujours évident, mais sachez qu'il est efficace et permet le chargement de fichier, mais il ne propose aucun habillage par défaut.

Il existe actuellement `Tk::Dressing` ([Lien 86](#)) qui propose quelques thèmes (ou plutôt habillages) par défaut. Vous devez l'installer pour l'utiliser. Il permet également une importation de votre propre thème en chargeant un fichier .ini dans lequel vous aurez mis les options voulues pour vos widgets. Ce fichier est de ce type.

```
[BrowseEntry]
-background: #A8A8A8
-foreground: #FFFFFF
-disabledbackground: #A8A8A8
-disabledforeground: #4E4E4E

[Button]
-activebackground: #686868
-background: #686868
-foreground: #FFFFFF
-disabledforeground: #414141
-activeforeground: #FFFFFF

[Canvas]
-background: #A8A8A8
```

Il est possible de proposer un thème au propriétaire du module afin qu'il puisse l'y intégrer. Ce module est simple d'utilisation et applique le thème choisi à un widget voulu (et ses enfants). Ainsi, vous pouvez avoir plusieurs widgets avec un habillage différent.

Voici un exemple d'utilisation du module issu de la documentation officielle.

```
#!/usr/bin/perl
use warnings;
use strict;
use Tk;
use Tk::Dressing;

use Tk::BrowseEntry;

my $TkDressing = Tk::Dressing->new();
```

```
my $mw = MainWindow->new( -title => "Dressing
widget", );
$mw->minsize( 300, 100 );

$mw->Button( -text => 'Close', )->pack(qw/ -side
bottom -padx 10 -pady 10 /);

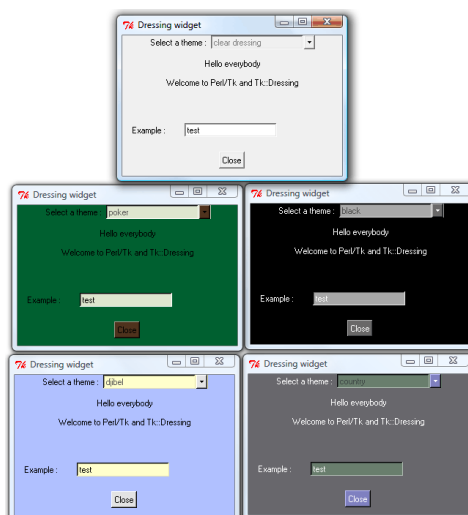
my $BrowseEntryTheme = $mw->BrowseEntry(
-label => "Select a theme : ",
-state => 'readonly',
-choices => [ 'clear dressing', sort
$TkDressing->get_all_theme ],
)->pack;

my $Message = "Hello everybody\n\nWelcome to
Perl/Tk and Tk::Dressing\n\n";
$mw->Label( -text => $Message, -anchor =>
'center' )->pack(qw/ -side top -padx 10 -pady
10 /);
$mw->Label( -text => 'Example : ', -anchor =>
'center' )->pack(qw/ -side left -padx 10 -pady 10
/);
$mw->Entry( -text => 'test', )->pack(qw/ -side
left -padx 10 -pady 10 /);

$BrowseEntryTheme->configure(
-browse2cmd => sub {
my $theme = $BrowseEntryTheme-
>Subwidget('entry')->get;
if ( $theme eq 'clear dressing' )
{ $TkDressing->clear($mw); return; }
$TkDressing->design_widget(
-widget => $mw,
-theme => $theme,
);
},
);

MainLoop();
```

Avec ce code, nous avons testé quatre thèmes fournis par le module. L'image ci-dessous montre l'habillage par défaut de la fenêtre Tk suivi des quatre thèmes.



On peut constater que l'on a un rendu complètement différent sans avoir à configurer à la main les différents widgets. N'hésitez pas à utiliser ce module !

Comment récupérer des fichiers ou répertoires via un drag & drop ?

Le but est de faire un **glisser-déposer** de fichiers ou répertoires dans un widget Text **sous Windows** afin de trouver le chemin et le nom du fichier ou répertoire.

```
#!/usr/bin/perl
use warnings;
use strict;
use Tk;
use Tk::Dropsite 'Win32';
use File::Basename;
use utf8;

my $mw = MainWindow->new( -background =>
'#E0F0FF', );

$mw->Label(
    -text      => 'Faire un glisser-déposer de
fichiers ou répertoires dans le widget ci-
dessous',
    -background => '#E0F0FF',
)->pack(qw/ -side top -pady 2/);

my $widgetTextDragDrop = $mw->Scrolled(
    'Text',
    -background => 'white',
    -scrollbars => 'osoe',
    -wrap       => 'none',
    -height     => 7,
)->pack(qw/ -side top -pady 2/);

$mw->Label(
    -text      => 'Résultat',
    -background => '#E0F0FF',
)->pack(qw/ -pady 2/);

my $widgetTextResultat = $mw->Scrolled(
    'Text',
    -background => 'white',
    -scrollbars => 'osoe',
    -wrap       => 'none',
)->pack(qw/ -pady 2 -fill both -expand 1/);

# Conception du glisser-deposer
$widgetTextDragDrop->DropSite(
    -dropcommand => [ \&ActionDrop,
$widgetTextDragDrop, $widgetTextResultat ],
    -droptypes  => 'Win32',
```

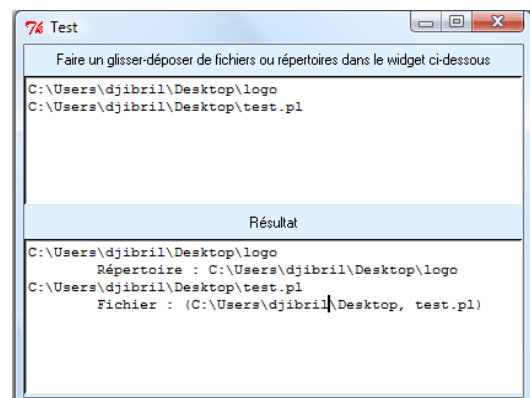
```
);

MainLoop;

sub ActionDrop {
    my ( $widgetTextDragDrop, $widgetTextResultat )
    = @_;

    my $sel = $widgetTextDragDrop->SelectionGet(
-selection => 'CLIPBOARD' );
    foreach my $fichier_rep ( split( /\n/, $sel ) )
    {
        $widgetTextDragDrop->insert( 'end',
"$fichier_rep\n" );
        $widgetTextResultat->insert( 'end',
"$fichier_rep\n" );
        if ( -d $fichier_rep ) {
            $widgetTextResultat->insert( 'end',
"\tRépertoire : $fichier_rep\n" );
        }
        elsif ( -f $fichier_rep ) {
            $widgetTextResultat->insert( 'end',
"\tFichier : (" . dirname($fichier_rep) .
', ' . basename($fichier_rep) . ") \n" );
        }
    }
    return;
}
```

Voici le résultat :



Pour appliquer le glisser-déposer au widget Text, on utilise la méthode **DropSite** du module **Tk::Dropsite** sur le widget Text. Puis pour effectuer les sélections, on utilise la méthode **SelectionGet** du module **Tk::Selection**.

Retrouvez la FAQ Perl/Tk en ligne : [Lien 87](#)



Qt : une meilleure intégration dans les prochaines versions d'Ubuntu

Des applications écrites en Qt pourraient être intégrées à l'installation par défaut

La distribution Linux Ubuntu devrait avoir un support amélioré des « Qt tools » dans sa prochaine version. C'est en tout cas ce qu'affirme son fondateur Mark Shuttleworth.

Pour préparer le terrain au déploiement futur d'Ubuntu sur un large panel de périphériques et de systèmes embarqués, Canonical travaille sur le binding de dconf pour Qt en collaboration avec l'expert en la matière Ryan Lortie.

L'objectif est que les applications Qt tournant sur les prochaines versions d'Ubuntu (après Natty Narwhal) utilisent le même framework que les autres applications Ubuntu, mettant donc Qt sur le même pied d'égalité que le reste de l'écosystème de la populaire distribution.

Les bibliothèques de Qt seront donc incluses dans le CD de Ubuntu. Shuttleworth affirme aussi que son équipe "va évaluer des applications développées en Qt en vue de leur inclusion dans le CD et dans l'installation par défaut de Ubuntu".

Cette adoption de Qt peut être vue par certains comme une autre critique à Gnome (et à Gtk donc), ce que Shuttleworth s'empresse de démentir sur son blog. Il s'agirait plutôt *"de la célébration de la diversité et la complexité du logiciel libre"*.

Gnome, l'environnement de bureau par défaut de longue date pour Ubuntu sera remplacé par « Unity » dans les prochaines versions de Ubuntu ([Lien 88](#)), en raison de différends entre Canonical et les développeurs de Gnome.

Pour rappel, Qt est un framework d'applications et d'interfaces utilisateurs multiplateforme qui permet aux développeurs d'écrire des applications qui fonctionnent aussi bien sur les environnements de bureau Windows, MacOS ou Linux mais aussi sur les plateformes mobiles et les systèmes intégrés.

Commentez cette news de Thibaut Cuvelier en ligne : [Lien 89](#)

La Freebox V6 développée avec Qt et QML

Un framework sera disponible pour créer ses propres applications

Je sais ça ne s'écrit pas comme ça, mais c'est une très bonne annonce que voilà pour les utilisateurs de Qt ! En effet, toute l'interface utilisateur de la Freebox V6 a été développée à l'aide de Qt et du langage déclaratif QML. Ce dernier peut être utilisé conjointement avec du code en JavaScript ou être intégré à une application C++.

Comme pour les PC et Mac, les développeurs auront plusieurs niveaux pour le développement d'applications : du natif et bas niveau, en C ou en C++ avec l'API 3D OpenGL ; un peu plus haut niveau avec Qt en C++ ou en JavaScript avec QML ; finalement, avec les standards du Web. Sans oublier que le framework de développement de jeux 2D en JavaScript, Elixir ([Lien 90](#)), sera toujours supporté.

Alors, elle n'est pas belle, la vie ?

Commentez cette news de dlewin en ligne : [Lien 91](#)

Ouverture de la rubrique PyQt & PySide

Le Python est un langage qui commence à prendre une certaine ampleur, que ce soit pour le développement Web avec le développement de frameworks comme Django ou pour le développement desktop. Ce dernier requiert principalement des interfaces graphiques (GUI), c'est pourquoi Python fournit de base Tkinter. Cependant, il n'est pas forcément le plus beau, le plus intuitif, le plus *user-friendly* - ni le plus *developer-friendly* - des frameworks d'interfaces pour Python. Du côté C++ de la planète, il se trouve que le framework Qt prend de plus en plus d'importance, notamment au niveau des interfaces graphiques.

C'est pourquoi, pour offrir plus qu'une alternative à Tkinter, RiverBanks a proposé PyQt, le premier binding Python de Qt, qui peut être utilisé avec à peu près toutes les versions de Qt (de la 1.43 à la 4.7.1), disponible sous licence GPL et commerciale.

Récemment, en août 2009, Nokia est entré en contact avec Riverbank pour arriver à un accord sur la disponibilité en LGPL de PyQt. L'objectif ? Démocratiser le développement avec Qt sur les plateformes mobiles, comme MeeGo. Cet accord n'a pas pu aboutir, d'où le lancement du projet PySide, le binding LGPL de Qt, notamment soutenu par Nokia. Ce projet arrive tout doucement à la version 1.0.0 finale.

D'où la création d'un forum ([Lien 92](#)), d'abord dénommé

PyQt, puis récemment élargi en PyQt et PySide. Le temps passant, la technologie a pris de l'essor, un certain nombre de rédacteurs nous ont proposé des tutoriels ([Lien 93](#)), des sources ([Lien 94](#)) ; Mark Summerfield, grand nom de Qt, a même rédigé un livre ([Lien 95](#)). Il était temps de rassembler toutes ces ressources francophones autour d'une même rubrique : <http://pyqt.developpez.com/> ([Lien 96](#))

Vous utilisez PyQt ou PySide ou un autre binding Python

de Qt ? Vous aimeriez participer à cette rubrique ? N'hésitez pas, tout ou presque reste encore à faire sur cette rubrique ! Traductions, tutoriels, cours, sources, entrées pour la FAQ ou toute autre folle idée qui vous passe par la tête, n'hésitez pas à nous en parler.

Bonne visite sur **la rubrique PyQt & PySide !**

Commentez cette news de Thibaut Cuvelier en ligne : [Lien 97](#)

Les derniers tutoriels et articles

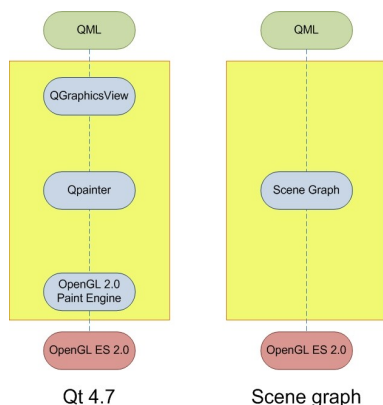
Roadmap de Qt

Qt Dev Days 2010 à Munich : le reportage

Sebastian Nyström est également l'orateur de la conférence "Qt Roadmap". Il commence par présenter les retours du marché avec ce que les utilisateurs attendent du framework Qt :

1. amélioration des performances : cela permet d'améliorer l'expérience utilisateur dans le cas de grosses applications en gardant l'interface fluide et prête à répondre ;
2. une intégration du Web sur tous les périphériques : Internet est maintenant partout et utilisé dans la majorité des applications ;
3. une amélioration des capteurs : les capteurs sont essentiels pour les applications embarquées et une amélioration de l'intégration des données est attendue ;
4. la 3D : pouvoir intégrer de plus en plus d'éléments en 3D et plus facilement dans des applications Qt ;
5. les écrans gérant plusieurs points de toucher : de plus en plus de périphériques utilisent le toucher pour contrôler les applications, il est important que Qt améliore la gestion du "multi-touch".

Le projet Scene Graph (graphe de scène) est l'un des projets sur lequel a travaillé l'équipe Qt en vue d'améliorer les performances entre QML et OpenGL (ES) 2.0. Le schéma ci-dessous présente les différentes étapes qui sont nécessaires dans Qt 4.7 et qui sont remplacées par ce nouveau graphe de scène permettant d'avoir un grand gain de performance. Le projet Scene Graph ne fait actuellement pas partie de Qt 4.7 mais sera intégré à la future version 4.8.



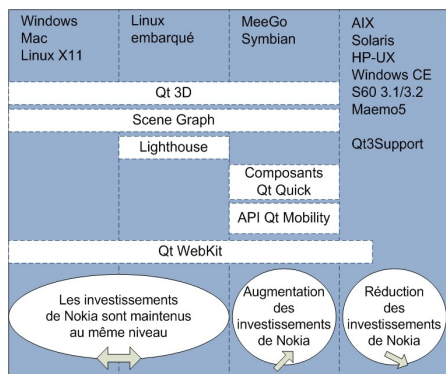
Le projet Lighthouse fait également partie de ce qui va améliorer considérablement la performance de Qt dans le domaine de l'accélération graphique sur périphériques embarqués. Il s'agit de quelque chose qui n'est pas directement perçu par les développeurs mais qui va permettre un gain de performance important même sur les plateformes à venir. Une conférence a été dédiée à ce projet ([Lien 98](#)).

QtWebKit est l'un des modules qui avancent le plus rapidement étant donné que les mises à jour sont indépendantes du framework Qt. Il est primordial pour les développeurs qui veulent intégrer des pages Web dans leur applications. QtWebKit s'adapte au fur et à mesure à l'avancement des spécifications de HTML5 et de CSS3 et pourra donc être utilisé pour afficher des pages écrites dans ces langages. L'équipe de QtWebKit travaille également sur le déplacement par toucher sur du contenu Web.

De nouvelles API sont également disponibles avec Qt Mobility 1.1 permettant de manipuler des données de périphériques mobiles, des capteurs et également d'utiliser des services intégrés. Vous pouvez également retrouver la liste des API sur l'annonce de leur sortie ([Lien 99](#)).

Qt/3D est une interface C++ portable et facile à utiliser pour créer des éléments graphiques en 3D. Qt/3D peut bien sûr être utilisé avec QML. Il s'agit d'une fonctionnalité qui sera peut-être intégrée à Qt 4.8 dans son propre module.

L'une des questions qui reviennent le plus souvent est : quel est l'investissement de Nokia sur les différentes plateformes que Qt supporte ? Le schéma ci-dessous décrit clairement quels sont les futurs investissements de Nokia et permet de voir qu'ils sont ciblés principalement sur les périphériques mobiles (Symbian et MeeGo) ainsi que QML et l'ensemble des projets présentés ci-dessus (Qt 3D, Scene Graph, Qt Mobility, Qt WebKit).



Cette présentation du futur de Qt est suivie d'une introduction à Qt Quick par Lars Knoll qui sera bientôt disponible en vidéo sur le site de Nokia.



Retrouvez la roadmap de Qt sur le reportage des Qt Dev Days 2010 à Munich en ligne : [Lien 100](#)

Qt Graphics et performance - velours et QML Scene Graph

Les animations devraient plus ressembler à du velours : une douce homogénéité plaisante. Du point de vue technique, cela requiert quelques petites choses.

Cet article est une traduction autorisée de Velvet and the QML Scene Graph de Gunnar ([Lien 101](#)).

1. L'article original

Le site Qt Labs permet aux développeurs de Qt de présenter les projets, plus ou moins avancés, sur lesquels ils travaillent ([Lien 102](#)).

Nokia, Qt, Qt Quarterly et leurs logos sont des marques déposées de Nokia Corporation en Finlande et/ou dans les autres pays. Les autres marques déposées sont détenues par leurs propriétaires respectifs.

Cet article est la traduction de l'article Velvet and the QML Scene Graph ([Lien 101](#)) de Gunnar ([Lien 103](#)) paru dans Qt Labs.

Cet article est une traduction d'un des tutoriels écrits par **Nokia Corporation and/or its subsidiary(-ies)**. Les éventuels problèmes résultant d'une mauvaise traduction ne sont pas imputables à Nokia.

2. Introduction

Tout d'abord, un peu de clarification. Aux DevDays cette année, beaucoup de personnes ont remarqué que le nom du projet, *scene graph*, était mal choisi. À cause d'une similarité dans le nom, il donnait l'impression de similarité avec des projets comme Open Scene Graph ([Lien 104](#)), ce qui n'est ni le cas, ni l'intention. Notre graphe de scène est un graphe 2D de petite taille et assez compact pour le rendu de fichiers XML. Ainsi, dès maintenant, nous y ferons référence comme au "QML Scene Graph". Son seul but dans l'existence est de rendre QML encore meilleur.

3. Les animations

Les animations devraient plus ressembler à du velours : une douce homogénéité plaisante. Du point de vue technique, cela requiert quelques petites choses.

Dessiner une image chaque fois que l'affichage le

permet. Des moniteurs modernes, comme les écrans LCD ou LED que vous utilisez pour lire ceci, sont pour la plupart réglés sur 60 Hz. Cela dépend de leur résolution (dpi), bien évidemment, mais des choses assez magiques peuvent avoir lieu aux alentours de 60 Hz. Si on met à jour des graphiques 2D à 30 Hz (soit trente fois par seconde), on peut voir chaque image comme une image individuelle. Dès qu'on se rapproche de 60 Hz, elles commencent à fusionner et les yeux perçoivent un mouvement fluide au lieu d'images juxtaposées, voilà la grande différence entre le velours et du papier de sable.

Être prête à temps. Pour atteindre 60 Hz, on doit calculer chaque image en au plus 16,66 millisecondes. Si on dépasse ce temps, on a raté le coche. Ce qui signifie que, pendant des animations, on ne peut pas faire autre chose que mettre à jour quelques propriétés puis dessiner le tout. Si ça prend du temps, cela nécessite soit d'être prêt au préalable, soit d'être fait dans un thread en arrière-plan, comme décrit dans l'article précédent. Jusqu'il y a peu, j'étais convaincu que rater le coche l'une ou l'autre fois ne serait pas désastreux, mais ça fait vraiment une différence, comme entre le velours et le papier de sable.

Ne pas dessiner en plein milieu d'un rafraîchissement vertical, cela mène à une déchirure d'écran, soit, d'une manière simple, votre image est découpée en deux, ce qui ruine ce qui aurait dû être un pur moment de plaisir visuel. La solution est d'utiliser une forme de synchronisation qui lie votre application et le taux de rafraîchissement vertical de l'écran, Qt ne vous y aide pas toujours. Sur Mac OS X et Symbian N8, le système vous limite à une fréquence de rafraîchissement vertical ; toute image que l'on tente de dessiner en plus provoque un blocage du thread de rendu par le système de fenêtrage. Sur Linux/X11, Maemo et Windows, il n'y a aucun blocage, ce phénomène peut donc apparaître. Heureusement, il y a une solution relativement simple. QGLWidget combiné à

QGLFormat::setSwapInterval mis à 1 autorisera à ce widget la synchronisation verticale. Dans le QML Scene Graph, nous requérons OpenGL pour définir l'intervalle d'échange à 1 par défaut.

L'avance de l'animation, relativement au temps entre chaque image. Si votre objet bouge à la vitesse d'un pixel par milliseconde, alors il doit bouger de 16,66 pixels par image. Si une image n'est pas dessinée, alors il doit bouger de 33,33 pixels la prochaine fois. Évidemment, si vous atteignez toujours l'objectif des 16,66 ms, ce problème n'en est pas un.

4. Et pourtant...

Même avec toutes ces précautions, Qt ne vous garantit pas encore le velours.

L'ingrédient manquant est l'endroit d'où le déclenchement de l'animation vient. Le framework Animation de Qt utilise un timer, qui avance les animations et met à jour les propriétés de tous les objets. Il envoie plusieurs requêtes de mise à jour et, finalement, l'image est redessinée. Il y a deux problèmes dans ce système. D'abord, le timer se déclenche toutes les 1000 / 60 ms, ce qui est arrondi à 16, ce qui n'est pas 16,66 comme cela aurait dû l'être. Ensuite, le timer n'est pas précis. Il peut se déclencher à 15 comme à 17 ; s'il se déclenche à 17, l'animation aura raté le coche une fois, une image ne sera pas dessinée.

Ceci m'a inquiété pendant un certain temps ; ainsi, pendant les DevDays de San Francisco, j'ai eu un peu de temps pour creuser le sujet. Le résultat est que le QML Scene Graph conduira les animations d'une manière un peu différente.

```
while (animationIsRunning) {
    processEvents();
    advanceAnimations();
    paintQMLScene();
    swapAndBlockForNextVSync();
}
```

Le résultat est qu'on progresse toujours dans l'animation en synchronisation avec le rafraîchissement vertical ; ainsi, on dessine une image toutes les 16,66 ms, soit très exactement à chaque fois qu'on peut en afficher une. J'ai dit que je n'étais pas un convaincu de la première heure, qu'on pouvait rater de temps en temps une image. Un seul coup d'œil au résultat. J'ai su qu'on y était. Le velours.

Nous ne pouvons pas faire ça généralement dans Qt, car cette méthode pour la synchronisation verticale ne se fait que par la fonction swapBuffers() d'OpenGL. On ne peut l'utiliser que pour une seule fenêtre. Avec Wayland ou des extensions personnalisées d'OpenGL, on pourrait potentiellement avoir la synchronisation verticale sans devoir échanger, ce qui signifie que, en théorie, on pourrait avancer dans les animations sur plusieurs fenêtres, mais c'est hors de portée pour le moment. Actuellement, c'est fixé pour une seule fenêtre utilisant le QML Scene Graph.

5. Conclusion

Le repository est ouvert. : [Lien 105](#)

Au nom de toute l'équipe Qt, j'aimerais adresser le plus grand remerciement à Nokia pour nous avoir autorisé la traduction de cet article !

Retrouvez l'article de Gunnar traduit par Thibaut Cuvelier en ligne : [Lien 106](#)

Génération de bindings PySide avec Shiboken

Ce tutoriel décrit le processus de création de bindings avec Shiboken. Une bibliothèque très simple et basée sur Qt sera utilisée en exemple.

1. L'article original

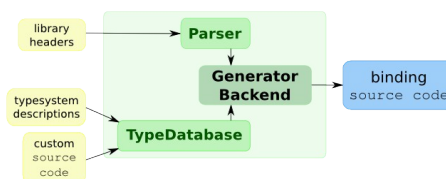
Le Qt Developer Network est un réseau de développeurs utilisant Qt afin de partager leur savoir sur ce framework. Vous pouvez le consulter en anglais ([Lien 107](#)).

Nokia, Qt, Qt Quarterly et leurs logos sont des marques déposées de *Nokia Corporation* en Finlande et/ou dans les autres pays. Les autres marques déposées sont détenues par leurs propriétaires respectifs.

Cet article est la traduction de PySide Binding Generation Tutorial ([Lien 108](#)).

2. Introduction

Cette image montre les entrées nécessaires pour générer le code source du binding :



L'utilisateur fournit les en-têtes de la bibliothèque avec un fichier de description du système de types montrant comment les classes seront exportées vers le langage cible, tout comme tout code source requis est à inclure dans les sources générées. Si aucune grosse modification n'est requise, ceci sera juste une liste déclarative de classes, d'espaces de noms, d'énumérations et de fonctions globales.

Ce tutoriel va passer en revue toutes les étapes nécessaires pour avoir un binding prêt à l'importation et à l'utilisation dans un programme Python. Le code est disponible ([Lien 109](#)). Il y a deux versions de l'exemple, le code source est le même, mais l'une utilise un Makefile général et l'autre des fichiers CMake ; la dernière est le système

utilisé par Shiboken et PySide.

3. La bibliothèque

Est présentée ici la petite et simpliste bibliothèque qui sera utilisée pour la suite de l'article, ainsi que les instructions pour la compiler. Il n'y a qu'une classe C++ dans cette bibliothèque, donc deux fichiers, ainsi qu'un fichier de projet pour la compilation avec QMake.

Les fichiers suivants seront considérés dans le dossier libfoo ; si vous les changez de place, étant donné que le Makefile bientôt présenté utilise ce répertoire en dur, vous rencontrerez quelques problèmes.

```
libfoo/foo.h
#ifndef FOO_H
#define FOO_H

#include <QtCore/QtCore>

class Math : public QObject
{
    Q_OBJECT
public:
    Math() {}
    virtual ~Math() {}
    int squared(int x);
};
#endif // FOO_H
```

```
libfoo/foo.cpp
#include "foo.h"

int Math::squared(int x)
{
    return x * x;
}
```

```
libfoo/foo.pro
TEMPLATE = lib
TARGET = foo
DEPENDPATH += .
INCLUDEPATH += .
HEADERS += foo.h
SOURCES += foo.cpp
```

Pour compiler :

```
cd libfoo
qmake
make
```

4. Le binding libfoo avec Shiboken

Pour mener à bien la création du binding, quelques outils doivent être installés sur le système :

1. Qt (avec les en-têtes et les fichiers .pc de pkg-config, ce qui correspond à un paquet -dev pour les distributions Debian) ;
2. PySide, le binding Python de Qt réalisé à l'aide de Shiboken ;
3. les systèmes de type pour les bindings Python de Qt ;
4. les en-têtes pour la bibliothèque à binder.

Avec ces outils, le développeur souhaitant réaliser un binding Python doit récolter ces informations, utilisées par le générateur pour le binding :

1. un fichier de description de système de type, pour donner la manière dont le binding doit être fait ;
2. global.h, un en-tête incluant tous les en-têtes de libfoo et définissant les macros requises ;
3. un système de compilation pour diriger le processus de génération, compilation et liaison.

4.1. Compilation avec Makefile

La structure des répertoires et le contenu pour le projet de binding par Makefile pourraient ressembler à ceci :

```
foobinding-makefile/
|-- foo/
|-- global.h
|-- Makefile
|-- pyside_global.h
`-- typesystem_foo.xml
```

Le répertoire foobinding-makefile devrait contenir le fichier global.h, le point central incluant tous les en-têtes de libfoo (dans ce cas, il appert que la bibliothèque n'a qu'un en-tête ; dans une bibliothèque plus réaliste, cependant, il devrait y en avoir beaucoup plus). Il pourrait aussi contenir les définitions de macros qui vont influencer le passage des en-têtes par le générateur. Le fichier pyside_global.h sera expliqué plus tard. Comme dit précédemment, le fichier typesystem_foo.xml décrit la manière dont l'export de la bibliothèque C++ se fera en Python.

Le répertoire foo est l'endroit où les sources générées seront placées. Au départ, il est vide et son nom sera le même que le nom de paquet trouvé dans le fichier de système de type :

```
<typesystem package="foo">
```

S'il y a besoin de code source écrit à la main de plus de deux lignes (ce qui le rend impropre à l'inclusion dans le fichier XML de système de type), il devrait être placé dans un répertoire glue. Pour ce binding, aucun code de ce type n'est requis.

Lors de l'écriture du fichier de système de type, il n'y a pas besoin de faire référence aux autres fichiers de système de type avec des chemins absolus, les endroits où ils peuvent être trouvés devraient être passés en paramètre au générateur (--typesystem-paths=PATH1:PATH2:[...]) ou dans la variable d'environnement TYPESYSTEMPATH.

4.2. Compilation avec CMake

L'arborescence est alors un peu plus élaborée :

```
foobinding-cmake/
|-- CMakeLists.txt
|-- foo/
|   |-- CMakeLists.txt
|   |-- global.h
|   |-- pyside_global.h
|   `-- typesystem_foo.xml
`-- tests/
    |-- CMakeLists.txt
    `-- math_test.py
```

Cette structure suit celle utilisée par PySide : un répertoire racine, un répertoire pour les informations sur le binding (comme dans l'exemple du Makefile) et un répertoire tests, un *must-have*.

5. La description de système de type

Le système de type est une spécification utilisée lors du mappage d'une bibliothèque C++ et du module Python correspondant. La spécification est un fichier XML écrit à la main, listant les types qui seront disponibles dans le binding généré, les modifications dans les signatures des classes et fonctions pour mieux correspondre au langage cible et la liste des composants qui devraient être rejetés du binding.

PySide utilise un format similaire à celui utilisé par Qt Jambi et Qt Script, déjà abondamment abordé dans la documentation de Qt Jambi ([Lien 110](#)).

5.1. Le cas de libfoo

Tout fichier de système de type débute avec la balise racine typesystem. L'attribut package contient le nom du paquet tel qu'il sera vu depuis Python.

Juste après, tous les fichiers fournissant les informations requises pour le processus de génération du binding en cours sont inclus, à la manière des en-têtes en C.

```
<?xml version="1.0"?>
<typesystem package="foo">
  <load-typesystem name="typesystem_core.xml"
generate="no" />
  <object-type name="Math" />
</typesystem>
```

Si le binding créé dépend d'autres bindings (dans ce cas-ci, libfoo dépend de QtCore, le binding de libfoo dépendra du binding de QtCore), le fichier nouvellement créé de système de type doit se référer aux fichiers de système de type de la bibliothèque dont il dépend.

L'inclusion d'autres fichiers de système de type se fait avec la balise load-typesystem. L'attribut generate doit être mis à no, sinon le générateur va tenter de recréer du code source pour les bindings déjà existants. Ces fichiers sont généralement situés dans /usr/share/PySide/typesystem, l'endroit précis peut être vérifié avec pkg-config :

```
pkg-config pyside --variable=typesystemdir
```

Maintenant, un mot d'explication. Les classes C++ peuvent être déclarées dans la description comme des types d'objet ou de valeur. La différence est subtile et tient plus du C++ que du Python. Si une classe C++ a un constructeur de copie public et un opérateur d'attribution, si elle peut être passée par valeur (comme QRect, QPoint, QString), si deux instances d'un objet peuvent être comparées et décrétées égales en se basant sur la valeur qu'elles contiennent (comme deux QPoint différents avec les mêmes coordonnées x et y), alors elle doit être déclarée comme un type de valeur. Si une classe C++ a un constructeur par copie et un opérateur d'attribution privés, si elle est prévue pour être passée comme pointeur (comme les QObject et dérivés), si les instances de cette

classe sont considérées comme uniques et égales uniquement à elles-mêmes, alors elle doit être déclarée comme objet de type.

Pour ce binding d'exemple, spécifier uniquement le nom de la classe suffit, puisque le système de génération va automatiquement repérer les méthodes avec des arguments et retourner la valeur de types connus. Ces types peuvent être décrits dans le même fichier ou dans un référencé par le biais de la balise load-typesystem.

Dans des situations plus complexes, les signatures des méthodes peuvent être changées ou rejetées avec d'autres balises, que vous pouvez consulter dans la référence ([Lien 111](#)).

5.2. Autres cas communs

Suivent maintenant des utilisations communes des capacités du système de type. Toutes sont utilisées dans les fichiers de système de type de PySide. Elles ne sont pas utilisées dans ce tutoriel.

5.2.1. Templates

Pour simplifier le processus d'écriture de code personnalisé pour le binding, des morceaux récurrents de code peuvent devenir plus génériques avec le mécanisme de template. Ils sont déclarés de cette manière :

```
<template name="only_bool*_fix">
  bool ok;
  %RETURN_TYPE retval = self.
%FUNCTION_NAME(&ok);
</template>
```

Et utilisés comme ceci :

```
<inject-code class="native" position="beginning">
  <insert-template name="only_bool*_fix"/>
</inject-code>
```

Le fichier typesystem_template.xml des bindings Qt peut être utilisé comme une bonne ressource pour ce genre d'exemples. Regardez aussi du côté de la documentation de Qt Jambi sur les templates ([Lien 112](#)).

5.2.2. Les classes ne dérivant pas de QObject

Même dans une bibliothèque basée sur Qt, il est assez fréquent de rencontrer des classes qui n'appartiennent pas à la hiérarchie QObject, elles doivent être déclarées comme ceci :

```
<value-type name="RectOrSomethingLikeThat"/>
```

5.2.3. Les fonctions globales

Les fonctions globales doivent être déclarées avec la balise function et identifiées avec leur signature :

```
<function signature="functionName(int, const
Object&); />
```

6. L'en-tête global

En plus des informations fournies par le système de type,

le générateur doit récolter des informations supplémentaires des en-têtes de la bibliothèque contenant les classes à exporter en Python. S'il y a un en-tête qui inclut tous les autres, il peut être passé directement au générateur.

Si un tel fichier n'est pas disponible, si seulement une partie de la bibliothèque est à convertir, si certains drapeaux (flags) doivent être déclarés à l'aide d'instructions `#define` avant le passage des en-têtes de la bibliothèque, alors un fichier `global.h` doit être fourni. Par exemple, si `NULL` n'est pas défini et est utilisé comme paramètre par défaut pour certains constructeurs ou méthodes, le parseur ne va pas le reconnaître.

```
#undef QT_NO_STL
#undef QT_NO_STL_WCHAR

#ifdef NULL
#define NULL 0
#endif

#include "pyside_global.h"
#include <foo.h>
```

`pyside_global.h` contient des inclusions et des définitions requises pour générer les bindings qui composent PySide. Il s'agit d'une copie du fichier résultant du traitement du fichier `pyside/PySide/global.pc.in`. La situation idéale serait d'avoir ce fichier fourni par PySide. Pour le moment, on se satisfera d'une copie dans les projets basés sur PySide.

7. Compiler le binding

Comme mentionné précédemment, le système de compilation utilisé doit effectuer ces différentes tâches dans le bon ordre :

- récolter des informations sur les endroits des en-têtes et des systèmes de type requis d'autres projets ;
- lancer le générateur avec les bons paramètres ;
- compiler et lier le binding.

7.1. Lancer le générateur

Le générateur est appelé avec les paramètres et options suivants :

```
generatorrunner --generatorSet=shiboken \
                global_header.h \
                --include-paths=$
(PATHS_TO_HEADERS)) \
                --typesystem-paths=$
(PATHS_TO_TYPESYSTEMS) \
                --output-directory=. \
                typesystem.xml
```

Notez que les variables pour l'inclusion et les systèmes de type auraient pu être déterminés au moment de la compilation avec l'outil `pkg-config` ou avec les informations fournies par les fichiers de configuration de CMake.

7.1.1. Récolter des informations avec `pkg-config`

Les bindings de Qt utilisent des informations pour la

compilation à travers le mécanisme `pkg-config`. Le nom des bindings Python de Qt est `pyside` et un simple `pkg-config pyside --cflags --libs` récupérera les informations requises pour la compilation du nouveau binding.

Le fichier `pyside.pc` des bindings Qt à destination de `pkg-config` requiert les fichiers `.pc` de Qt. Si la bibliothèque est à un endroit inhabituel comme `/opt/qt47`, il faut exporter la variable d'environnement `PKG_CONFIG_PATH` :

```
export
PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/opt/qt47/lib/pk
gconfig
```

Ces informations sont aussi disponibles grâce à `pkg-config` et à la variable `typesystemdir`. On l'utilise comme ceci :

```
pkg-config pyside --variable=typesystemdir
```

Ceci fournit des informations sur la localisation des fichiers de système de type utilisés pour les bindings Qt. Comme dit précédemment, le binding en cours de création les requiert en complément de ses propres informations pour le processus de génération.

Des informations en droite ligne du générateur de binding Shiboken sont aussi requises pour la compilation, son nom `pkg-config` est `shiboken`.

7.1.2. Récolter des informations avec CMake

Lors de la compilation de votre binding avec CMake, les informations nécessaires peuvent être incluses depuis le fichier `CMakeLists.txt` de votre projet en utilisant :

```
find_package(Shiboken REQUIRED)
find_package(PySide REQUIRED)
```

L'inclusion obligatoire des paquets Shiboken et PySide va définir un certain nombre de variables, selon le fichier `PySideConfig.cmake` :

- `PYSIDE_INCLUDE_DIR` : les répertoires à inclure pour utiliser PySide ;
- `PYSIDE_LIBRARY` : les fichiers à lier pour utiliser PySide ;
- `PYSIDE_PYTHONPATH` : l'endroit où les modules Python de PySide peuvent être trouvés ;
- `PYSIDE_TYPESYSTEMS` : les fichiers de système de type qui devraient être utilisés par les bindings étendant PySide.

De même, le fichier `ShibokenConfig.cmake` fournit d'autres informations requises :

- `SHIBOKEN_INCLUDE_DIR` : les répertoires à inclure pour utiliser Shiboken ;
- `SHIBOKEN_LIBRARIES` : les fichiers à lier pour utiliser Shiboken ;
- `SHIBOKEN_BUILD_TYPE` : précise si Shiboken a été compilé en Release ou en Debug ;
- `SHIBOKEN_PYTHON_INTERPRETER` : interpréteur Python (Release ou Debug) à utiliser avec les bindings ;
- `SHIBOKEN_PYTHON_LIBRARIES` : les bibliothèques Python (Release ou Debug) auxquelles Shiboken est lié.

7.2. Compiler

Cette section présente les deux méthodes : Makefile et CMake.

7.2.1. Makefile

Voici un Makefile à utiliser :

```
LIBFOO_DIR = `pwd`/../../libfoo
LIBS = `pkg-config pyside --libs` \
      -L$(LIBFOO_DIR) -lfoo
CXXFLAGS = -I/usr/share/qt4/mkspecs/linux-g++ -I. \
          -I$(LIBFOO_DIR) \
          -I`pwd`/foo \
          -I`pkg-config --variable=includedir \
pyside`/QtCore/ \
          -I`pkg-config --variable=includedir QtCore` \
          -I`pkg-config --variable=includedir QtCore`/.. \
          -I`pkg-config --variable=includedir QtGui` \
          `pkg-config pyside --cflags`

QT4SYSTEM_DIR = `pkg-config pyside \
--variable=typesystemdir`
QT4HEADER_DIRS = `pkg-config \
--variable=includedir QtCore`:`pkg-config \
--variable=includedir QtCore`/..
PYSIDE_PYTHONPATH = `pkg-config \
--variable=pythonpath PySide`
PYTHON_INTERPRETER = `pkg-config \
--variable=python_interpreter shiboken`

all: generate compile link

generate:
    generatortrunner --generatorSet=shiboken \
    global.h \
    --include-paths=$(LIBFOO_DIR):$ \
    (QT4HEADER_DIRS):/usr/include \
    --typesystem-paths=.:$(QT4SYSTEM_DIR) \
    --output-directory=. \
    typesystem_foo.xml

compile:
    g++ foo/foo_module_wrapper.cpp \
    foo/math_wrapper.cpp -Wall -fPIC $(CXXFLAGS) -c

link:
    g++ foo_module_wrapper.o math_wrapper.o $(LIBS) \
    -fPIC -shared -Wl,-soname,foo.so -o foo.so

test:
    LD_LIBRARY_PATH=$(LIBFOO_DIR):$(LD_LIBRARY_PATH) \
    PYTHONPATH=$(PYSIDE_PYTHONPATH):$(PYTHONPATH) \
    $(PYTHON_INTERPRETER) -c \
    "import foo; m = foo.Math(); print '5 squared \
is %d' % m.squared(5)"

clean:
    rm -rf *.o *.so *.?pp *.log *.log foo/*
```

Gardez en tête que ce Makefile attend que les répertoires libfoo et foobinding-makefile soient au même niveau dans l'arborescence des dossiers. Il faut adapter le code si vous avez fait autrement.

Maintenant, compilez et liez le binding avec make :

```
cd foobinding-makefile
make
make test
```

La commande make test va lancer l'interpréteur Python avec ce bout de code, qui va importer le module du binding, en instancier une classe, lancer une méthode et afficher le résultat :

```
import foo; m = foo.Math(); print '5 squared is \
%d' % m.squared(5)
```

7.2.2. CMake

Voici un fichier CMakeLists.txt à utiliser :

```
project(foobinding)

cmake_minimum_required(VERSION 2.6)

find_package(PythonLibs REQUIRED)
find_package(Shiboken REQUIRED)
find_package(PySide REQUIRED)
find_package(Qt4 4.6.2 REQUIRED)

set(LIBFOO_DIR ${CMAKE_SOURCE_DIR}/../../libfoo)

find_program(GENERATOR generatortrunner REQUIRED)
if (NOT GENERATOR)
    message(FATAL_ERROR "You need to specify \
GENERATOR variable (-DGENERATOR=value)")
endif()

if(CMAKE_HOST_UNIX)
    option(ENABLE_GCC_OPTIMIZATION "Enable \
specific GCC flags to optimization library size \
and performance. Only available on Release Mode" \
0)
    set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -Wall \
-fvisibility=hidden -Wno-strict-aliasing")
    set(CMAKE_CXX_FLAGS_DEBUG "-g")
    if(ENABLE_GCC_OPTIMIZATION)
        set(CMAKE_BUILD_TYPE Release)
        set(CMAKE_CXX_FLAGS_RELEASE "-DNDEBUG -Os \
-Wl,-O1")
    else()
        set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} \
${CMAKE_CXX_FLAGS} -Wl,--hash-style=gnu")
    endif()
endif()

if(CMAKE_HOST_APPLE)
    if (NOT QT_INCLUDE_DIR)
        set(QT_INCLUDE_DIR \
"/Library/Frameworks")
    endif()
endif()

include(${QT_USE_FILE})

enable_testing()

add_subdirectory(foo)
add_subdirectory(tests)
```

Ceci est le CMakeLists.txt principal du projet, un fichier CMake tout à fait régulier ; en cas de doute, la documentation CMake est toujours disponible ([Lien 113](#)).

Remarquez que les tests ont été activés avec `enable_testing()`.

foobinding-cmake/foo/CMakeLists.txt

```
project(foo)

set(foo_SRC
    ${
    {CMAKE_CURRENT_BINARY_DIR}/foo/foo_module_wrapper
    .cpp
    }
    ${
    {CMAKE_CURRENT_BINARY_DIR}/foo/math_wrapper.cpp
    }
)

set(foo_INCLUDE_DIRECTORIES
    ${SHIBOKEN_INCLUDE_DIR}
    ${PYTHON_INCLUDE_PATH}
    ${PYSIDE_INCLUDE_DIR}
    ${PYSIDE_INCLUDE_DIR}/QtCore
    ${QT_INCLUDE_DIR}
    ${QT_QTCORE_INCLUDE_DIR}
    ${LIBFOO_DIR}
)

set(foo_LINK_LIBRARIES
    ${QT_QTCORE_LIBRARY}
    ${SHIBOKEN_PYTHON_LIBRARIES}
    ${SHIBOKEN_LIBRARY}
    ${PYSIDE_LIBRARY}
    ${LIBFOO_DIR}/libfoo.so
)

include_directories(foo ${
    {foo_INCLUDE_DIRECTORIES}
})
add_library(foo MODULE ${foo_SRC})
set_property(TARGET foo PROPERTY PREFIX "")
target_link_libraries(foo ${foo_LINK_LIBRARIES})

add_custom_command(OUTPUT ${foo_SRC}
    COMMAND ${GENERATOR}
    --generatorSet=shiboken
    --enable-parent-ctor-heuristic --enable-pyside-
    extensions --enable-return-value-heuristic
    $
    {CMAKE_SOURCE_DIR}/foo/global.h
    --include-paths=${
    {QT_INCLUDE_DIR}:${LIBFOO_DIR}
    }
    --typesystem-paths=${
    {typesystem_path}:${PYSIDE_TYPESYSTEMS}
    }
    --output-directory=${
    {CMAKE_CURRENT_BINARY_DIR}
    }
    $
    {CMAKE_CURRENT_SOURCE_DIR}/typesystem_foo.xml
    WORKING_DIRECTORY $
    {CMAKE_CURRENT_SOURCE_DIR}
    COMMENT "Running generator for
    libfoo..."
)
```

Ceci est le fichier CMakeLists.txt du répertoire du binding, la commande `add_custom_command` est responsable de l'appel du générateur Shiboken avec les paramètres et variables appropriés. Remarquez que les options de la ligne de commande sont directement liées aux particularités des bindings Qt ; pour un binding C++ pur, aucune n'aurait été nécessaire :

```
--enable-parent-ctor-heuristic --enable-pyside-
extensions --enable-return-value-heuristic
```

foobinding-cmake/tests/CMakeLists.txt

```
if(WIN32)
    set(TEST_PYTHONPATH "${foo_BINARY_DIR};$
    {PYSIDE_PYTHONPATH}")
    set(TEST_LIBRARY_PATH "${LIBFOO_DIR};
    $ENV{PATH}")
    set(LIBRARY_PATH_VAR "PATH")
    string(REPLACE "\\\" "/" TEST_PYTHONPATH "$
    {TEST_PYTHONPATH}")
    string(REPLACE "\\\" "/" TEST_LIBRARY_PATH "$
    {TEST_LIBRARY_PATH}")

    string(REPLACE ";" "\\;" TEST_PYTHONPATH "$
    {TEST_PYTHONPATH}")
    string(REPLACE ";" "\\;" TEST_LIBRARY_PATH "$
    {TEST_LIBRARY_PATH}")
else()
    set(TEST_PYTHONPATH "${foo_BINARY_DIR};$
    {PYSIDE_PYTHONPATH}")
    set(TEST_LIBRARY_PATH "${LIBFOO_DIR};
    $ENV{LD_LIBRARY_PATH}")
    set(LIBRARY_PATH_VAR "LD_LIBRARY_PATH")
endif()

add_test(math ${SHIBOKEN_PYTHON_INTERPRETER} $
    {CMAKE_CURRENT_SOURCE_DIR}/math_test.py)
set_tests_properties(math PROPERTIES ENVIRONMENT
    "PYTHONPATH=${TEST_PYTHONPATH};$
    {LIBRARY_PATH_VAR}=${TEST_LIBRARY_PATH}")
```

Ce fichier peu évolué informe CMake des tests à effectuer et avec quelles variables.

La meilleure chose à faire quand on compile avec CMake est de créer un répertoire de compilation et de le lancer depuis ce dernier (compilation hors sources).

```
cd foobinding-cmake
mkdir build
cd build
cmake ..
make
```

Finalement, un petit test unitaire très simple :

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

'''Test cases for foo bindings module.'''

import unittest
import foo

class MathTest(unittest.TestCase):

    def testMath(self):
        '''Test case for Math class from foo
        module.'''
        val = 5
        math = foo.Math()
        self.assertEqual(math.squared(5), 5 * 5)

if __name__ == '__main__':
    unittest.main()
```

Pour lancer le test :

```
ctest
```

La sortie devrait ressembler à ceci :


```
Test project YOURPATH/binding-
tutorial/foobinding-cmake/build
  Start 1: math
1/1 Test #1: math .....
Passed      0.10 sec

100% tests passed, 0 tests failed out of 1

Total Test time (real) =  0.11 sec
```

Pour plus de détails, utilisez le paramètre -V.

8. Conclusion

C'est à peu près tout. D'autres exemples de CMakeLists.txt et de tests unitaires sont disponibles dans les sources de PySide ([Lien 113](#)).

Retrouvez la traduction de Thibaut Cuvelier en ligne : [Lien 114](#)

Windows Phone

Les derniers tutoriels et articles



Bien commencer à développer pour Windows Phone 7

Cet article a pour but de présenter le développement d'application pour Windows Phone 7 en Silverlight à l'aide de MVVM Light

1. Introduction

Windows Phone 7 est le dernier système d'exploitation mobile proposé par Microsoft au mois d'octobre 2010. Microsoft propose deux types d'application pour son téléphone :

- **Silverlight** pour les applications classiques contenant des listes, champs texte, boutons, etc. ;
- **XNA** pour les applications plus graphiques (les jeux par exemple).

Les deux technologies ne sont pas incompatibles. En effet pour jouer un son dans une application Silverlight, il faudra utiliser la DLL **Microsoft.Framework.XNA**.

2. Le modèle MVVM

MVVM (Model View ViewModel) est un pattern conçu spécialement pour WPF et Silverlight. Ce nouveau pattern est parti du principe que si l'interface utilisateur (IHM) est fortement liée au reste du code de l'application, le designer en charge de l'IHM doit avoir des connaissances en développement. Or, ce type de profil est rare (donc cher). L'un des objectifs de ce pattern est ainsi de délier la vue du reste du code.

La **View** (vue) s'occupe uniquement de l'affichage des données et ne traite aucun événement ou traitement des données dans le code-behind. Le seul code autorisé dans le code-behind d'un fichier XAML doit simplement permettre de gérer des actions uniquement en relation avec la vue (animations par exemple).

Le **ViewModel** (modèle de la vue) est une abstraction de la vue. Il contient toutes les commandes et propriétés qui seront bindées dans la vue directement dans le XAML.

Le **Model** possède les objets métier dont vous aurez besoin dans votre application ainsi que la communication vers votre source de données (XML, base de données, services, etc.).

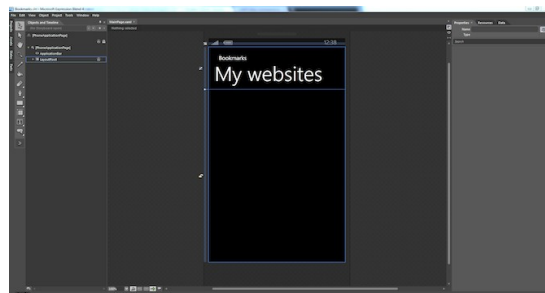


Les deux principaux avantages du MVVM sont :

- comme énoncé plus haut, la vue est totalement séparée du reste de l'application, elle peut être remplacée facilement sans perturber le fonctionnement de l'application ;
- cette séparation facilite également les tests sur chacune des parties.

3. Les outils nécessaires

Les applications Windows Phone 7 se développent sur l'IDE de **Microsoft Visual Studio 2010** ([Lien 115](#)) mais nécessitent l'installation des Windows Phone 7 Developer Tools ([Lien 116](#)).

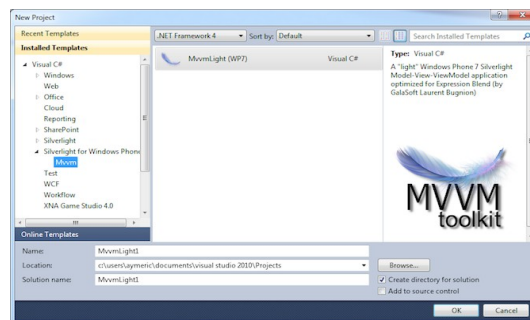


4. MVVM Light

MVVMLight est un ensemble d'outils créés par le MVP Suisse Laurent Bugnion permettant de mettre en place rapidement une architecture MVVM pour Silverlight.

L'installation de ce toolkit est décrite ici sur le site de l'auteur. Attention, vous l'utilisez pour Windows Phone 7 et devez donc installer un correctif pour que tout fonctionne parfaitement.

Pour démarrer un nouveau projet Windows Phone 7 en utilisant MVVMLight, créez un nouveau projet de type **Mvvm**.

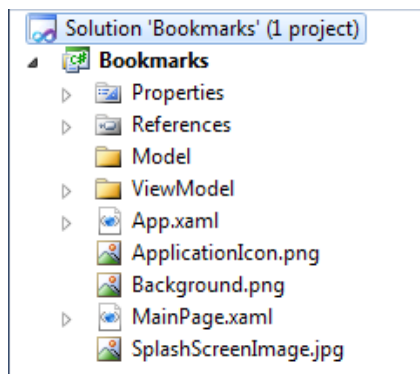


5. Votre première application

Dans cet article, je vous propose de développer une application qui affiche une liste de sites Internet. Lors du clic sur un des éléments de la liste, le navigateur du téléphone s'ouvre et affiche le site Web en question. Ce n'est pas l'application de l'année, mais son but est simplement d'exposer les principes du développement Silverlight sur Windows Phone 7.

- Créer un projet MVVMLight pour Windows Phone 7 comme énoncé dans la section précédente. Nous l'appellerons "Bookmarks" :

Avant d'aller plus loin, analysons les fichiers et dossiers déjà créés :



- **Model** : dossier qui contiendra les fichiers relatifs au modèle de l'application (classes métier et accès aux données) ;
- **ViewModel** : Dossier qui contiendra les ViewModels de l'application :

- **MainViewModel.cs** : ViewModel correspondant à la vue MainPage.xaml ;

- **ViewModelLocator.cs** :

pour délier encore un peu plus la View du ViewModel, le lien entre ces deux fichiers se fera par l'intermédiaire d'un **Locator**. La vue et le modèle de vue ne se connaissent donc pas directement. La vue demande au Locator une référence vers son modèle de vue. À chaque création d'un ViewModel, il faudra mettre à jour le fichier Locator ;

- **App.xaml** : fichier XAML qui fait référence aux ressources et qui gère les événements au niveau de l'application ;
- **ApplicationIcon.png** : icône de l'application ;
- **Background.png** : icône qui sera affichée lorsque l'application est "épinglée" (pinned) sur la page d'accueil du téléphone ;
- **MainPage.xaml** : page d'accueil de l'application (View) ;
- **SplashScreenImage.jpg** : splash screen de l'application (image affichée pendant le lancement de l'application).

Dans le code présenté ci-dessous, les commentaires ont été supprimés pour plus de lisibilité.

5.1. Création du Model

```
Website.cs
namespace Bookmarks.Model
{
    public class Website
    {
        public string Title { get; set; }
        public string Url { get; set; }
    }
}
```

```

    }
}

WebsitesList.cs
using System.Collections.ObjectModel;

namespace Bookmarks.Model
{
    public class WebsitesList :
        ObservableCollection<Website>
    {
        public WebsitesList() : base()
        {
            Add(new Website()
            {
                Title = "Aymeric's blog",
                Url = "http://www.aymericlagier.com"
            });
            Add(new Website()
            {
                Title = "Etudiant.ms",
                Url = "http://etudiant.ms"
            });
            Add(new Website()
            {
                Title = "Julien Dollon",
                Url = "http://blogs.dotnet-
france.com/juliend/"
            });
        }
    }
}
```

Explications

Cette classe hérite simplement de `ObservableCollection<T>` ([Lien 117](#)) et contient des objets de types *Website*. Ce sera la source de données de notre application.

5.2. Modification du ViewModel

```

using Bookmarks.Model;
using GalaSoft.MvvmLight;
using GalaSoft.MvvmLight.Command;
using Microsoft.Phone.Tasks;

namespace Bookmarks.ViewModel
{
    public class MainViewModel : ViewModelBase
    {
        public string ApplicationTitle { get { return
        "Bookmarks"; } }
        public string PageName { get { return "My
        websites"; } }

        public RelayCommand BrowseCommand {get; set;}

        private Website _SelectedItem;
        public Website SelectedItem
        {
            get { return _SelectedItem; }
            set { _SelectedItem = value; }
        }

        private WebsitesList _Websites;
```

```

public WebsitesList Websites
{
    get { return _Websites; }
    set { _Websites = value; }
}

public MainViewModel()
{
    if (IsInDesignMode)
    {
        // Code runs in Blend --> create design
time data.
    }
    else
    {
        Websites = new WebsitesList();
        BrowseCommand = new RelayCommand(Browse);
    }
}

private void Browse()
{
    if (SelectedItem != null)
    {
        WebBrowserTask wb = new WebBrowserTask();
        wb.URL = SelectedItem.Url;
        wb.Show();
    }
}
}

```

Explications

- *SelectedItem* correspond au **Website** sélectionné dans la vue via la listBox.
- **Websites** est un objet de type **WebsitesList** est sera la source de données de la listBox permettant d'afficher les objets de type **Website**.
- Le constructeur du ViewModel permet d'instancier la commande **BrowseCommand** en lui indiquant quelle méthode exécuter lors du clic sur un **Website**.
- La méthode **Browse** fait appel à **WebBrowserTask**, utilisé pour appeler le navigateur du téléphone.

5.3. Modification de la View

```

<phone:PhoneApplicationPage
x:Class="Bookmarks.MainPage"
xmlns="http://schemas.microsoft.com/winfx/2006/
xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/200
6/xaml"
xmlns:phone="clr-
namespace:Microsoft.Phone.Controls;assembly=Micro
soft.Phone"
xmlns:shell="clr-
namespace:Microsoft.Phone.Shell;assembly=Microsof
t.Phone"
xmlns:d="http://schemas.microsoft.com/expressio
n/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/mar
kup-compatibility/2006"
xmlns:cmd="clr-
namespace:GalaSoft.MvvmLight.Command;assembly=Gala
Soft.MvvmLight.Extras.WP7"
xmlns:i="clr-
namespace:System.Windows.Interactivity;assembly=S
ystem.Windows.Interactivity"

```

```

    FontFamily="{StaticResource
PhoneFontFamilyNormal}"
    FontSize="{StaticResource PhoneFontSizeNormal}"
    Foreground="{StaticResource
PhoneForegroundBrush}"
    SupportedOrientations="Portrait"
    Orientation="Portrait"
    mc:Ignorable="d"
    d:DesignWidth="480"
    d:DesignHeight="768"
    shell:SystemTray.IsVisible="True"
    DataContext="{Binding Main,
Source={StaticResource Locator}}">

    <!--LayoutRoot contains the root grid where all
other page content is placed-->
    <Grid x:Name="LayoutRoot"
        Background="Transparent">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto" />
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>

        <!--TitlePanel contains the name of the
application and page title-->
        <StackPanel x:Name="TitlePanel"
            Grid.Row="0"
            Margin="24,24,0,12">
            <TextBlock x:Name="ApplicationTitle"
                Text="{Binding
ApplicationTitle}"
                Style="{StaticResource
PhoneTextNormalStyle}" />
            <TextBlock x:Name="PageTitle"
                Text="{Binding PageName}"
                Margin="-3,-8,0,0"
                Style="{StaticResource
PhoneTextTitle1Style}" />
        </StackPanel>

        <!--ContentPanel - place additional content
here-->
        <Grid x:Name="ContentGrid" Grid.Row="1">
            <ListBox x:Name="MainListBox" Margin="0,0,-
12,0"
                ItemsSource="{Binding Websites}"
                SelectedItem="{Binding SelectedItem}">
                <i:Interaction.Triggers>
                    <i:EventTrigger
                        EventName="SelectionChanged">
                        <cmd:EventToCommand Command="{Binding
BrowseCommand}" />
                    </i:EventTrigger>
                </i:Interaction.Triggers>
            <ListBox.ItemTemplate>
                <DataTemplate>
                    <StackPanel Margin="0,0,0,17"
                        Width="432">
                        <TextBlock Text="{Binding Title}"
                            TextWrapping="Wrap"
                            Style="{StaticResource
PhoneTextExtraLargeStyle}" />
                    </StackPanel>
                </DataTemplate>
            </ListBox.ItemTemplate>
        </ListBox>
        </Grid>
    </Grid>
</phone:PhoneApplicationPage>

```

Explications

Cette page est composée de deux titres bindés sur les propriétés `ApplicationTitle` et `PageName` et d'une `ListBox` liée à la propriété `Websites` du `ViewModel`. `Websites` étant une collection de " `Website` ", l'élément courant sélectionné est bindé sur une propriété de type `Website` : `SelectedItem`.

Le trigger présent sur la `ListBox` permet de déclencher la commande `BrowseCommand` dès que l'événement `SelectionChanged` est déclenché. Cette commande est implémentée dans le `ViewModel`.

5.4. Publication sur le Marketplace

Pour que votre application soit disponible au grand public, il faut la publier sur le Marketplace via le AppHub.

Plusieurs blogueurs ont déjà traité de ce sujet, voici un résumé de ces articles avec leurs liens.

Si vous êtes étudiants, vous pouvez suivre l'article de Mathieu Perrein ([Lien 118](#)) (MSP) : [WP7] Créez votre compte Marketplace ! ([Lien 119](#))

En effet, les étudiants ne payent pas pour mettre leurs applications sur le Marketplace (économie de 99 \$ par an). En revanche, le processus d'inscription est un peu compliqué. Pour vérifier votre identité, Microsoft fait appel à une société appelée GeoTrust.

Comme vous ne payez pas à l'inscription, GeoTrust n'est pas averti de votre inscription et ne vous contacte donc pas pour vérifier votre identité. L'astuce est de poster une application vide sur le Marketplace afin que GeoTrust vous contacte. Le processus est expliqué par Mathieu dans son article.

Pour un processus d'inscription plus général, suivez l'article de Pierre Cauchois : [Windows Phone 7] Inscription sur Marketplace : les ressources pour s'en

sortir dans toutes les situations ([Lien 120](#)).

À savoir :

- Microsoft ne permet de publier que cinq applications gratuites sans occasionner de frais. À partir de la sixième application gratuite, vous devrez déboursier 19,99 \$ par application gratuite postée.
- Avant de commencer à développer, prenez soin de lire le Application Certification Requirements qui liste ce que vous devez respecter pour que votre application soit validée sur le Marketplace. Par exemple, votre code ne peut pas faire appel à `Microsoft.Xna.Framework.Game` s'il utilise `System.Windows.Control`. Si c'est le cas, la certification échouera et votre application ne sera pas publiée sur le Marketplace.
- Chaque application postée sur le Marketplace est testée et certifiée avant d'être publiée. L'objectif de Microsoft est de valider les applications dans les cinq jours ouvrés. Cet objectif est dans 90 % des cas respecté.

6. Liens

- MVVM et Silverlight 4 : [Lien 121](#)
- Section Windows Phone 7 de Frogz : [Lien 122](#)
- Blog de Pierre Cauchois : [Lien 123](#)
- MVVMLight : [Lien 124](#)
- Comment obtenir un numéro ITIN ? : [Lien 125](#)

7. Conclusion

Cet article ne rentre pas dans les détails du développement WP7 mais donne un premier aperçu des techniques de développement et de publication d'applications pour cette plateforme.

Le projet créé dans cet article est disponible ici : [Lien 126](#).

Retrouvez l'article d'Aymeric Lagier en ligne : [Lien 127](#)



Le Mac a son App Store

L'App Store d'Apple, intégré à iOS, est connu pour être la plateforme de téléchargement la plus complète et la plus utilisée. Elle contient plus de 300 000 applications téléchargées plus de trois milliards de fois sur quelque 70 millions d'iPhone, iPad et iPod Touch (chiffres évalués en octobre 2010).

Fort de ce succès impressionnant, Apple n'a pas souhaité s'arrêter là.

Dans quelques jours, vous pourrez découvrir le Mac App Store.

1. L'App Store : un outil apprécié

Au-delà des chiffres, ce qui plaît beaucoup dans l'App Store, c'est sa facilité d'utilisation.

La procédure d'installation d'une application est simple :

- on crée un compte iTunes ;
- on enregistre sa carte bancaire ;
- il suffit ensuite de sélectionner l'application à télécharger pour qu'elle s'installe automatiquement.

Pour ce qui est de la mise à jour, une alerte informe automatiquement du nombre de mises à jour en attente : il suffit de valider les mises à jour et le reste se fait tout seul.

Il en est de même pour le changement d'appareil : si vous avez acheté une application sur iPhone et que vous souhaitez la récupérer sur iPad, il vous suffit de rentrer vos coordonnées iTunes dans l'App Store pour la télécharger de nouveau gratuitement.

Pour résumer, cette solution est à la fois :

- simple ;
- rapide ;
- automatique ;
- économe (on achète une fois, on télécharge à souhait).

2. L'écran d'accueil iOS et ses applications

Ce qui fait le succès d'iOS, ce sont évidemment ses applications, qui se lancent instantanément, en plein écran et se stoppent tout aussi facilement.



Second facteur de ce succès, la facilité pour retrouver une application : l'écran d'accueil.

Toutes les icônes y sont accessibles facilement, rangées éventuellement dans des dossiers.

3. L'ordinateur personnel : un outil compliqué

Aujourd'hui, lorsque l'on a un ordinateur, les choses ne sont pas si simples que sur un iPhone ou un iPad.

Je veux aller sur Internet. Si je suis sur un PC Windows (Mac et Linux sont pour le moment épargnés), je dois m'assurer que j'ai bien installé mes antivirus, qu'ils sont à jour, que j'ai bien mon antispam, qu'il est à jour et que j'ai fait toutes les mises à jour nécessaires.

Là, je peux aller sur Internet.

Maintenant, j'ai besoin d'un logiciel de traitement de texte.

Lequel choisir ? Où le trouver ?

Combien va-t-il me coûter ?

Comment vais-je payer ?

Comment ça se passe pour les mises à jour ?

Est-ce le vrai ou est-ce un virus caché ?

Je caricature un peu, mais c'est quand même un peu ça.

J'ai tous mes logiciels installés, l'ordinateur tombe en panne, je perds mes licences, je dois télécharger la bonne version, compatible avec mon système.

Ensuite, je veux simplement aller sur Facebook ou Twitter, sans y passer trois jours.

Ponctuellement, rapidement. Ce n'est pas forcément si simple : il faut trouver le bon logiciel, parfois se rendre sur Internet...

Et encore, si on a trouvé le logiciel parmi les nombreuses applications installées (en particulier sous Windows, où il faut parcourir menus, sous-menus voire sous-sous-menus...)

En clair, j'ai un ordinateur super complet pour le boulot : ça c'est génial. Par contre, dès que je veux faire un truc simple et rapide, c'est impossible.

4. Mac App Store : simplifiez-vous l'informatique

Apple a su, par son **expérience iOS**, montrer son intérêt pour la **simplicité et l'efficacité**.



C'est d'ailleurs ce qui fait le succès du Mac, malgré sa part encore mince face au PC. Pour faire simple et rapide, le Mac représente **20,7 % des ventes d'ordinateurs aux États-Unis**. Au total, cela représente tout de même **50 millions d'utilisateurs dans le**

monde.

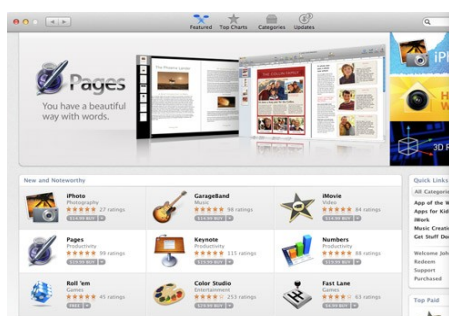
Aujourd'hui, Mac OS X est déjà très optimisé pour la simplicité et l'efficacité : un menu Applications rapide d'accès, pas de fioritures, une seule barre de menu, Spotlight, une ergonomie épurée. **Tout ce qui fait que, dans la majorité des cas, un utilisateur Mac ne reviendra jamais sur PC.**

Cependant, on n'est pas encore au niveau d'iOS mais en même temps, il faut conserver cet aspect « outil de travail » qui le rend aussi puissant que compliqué.

C'est donc là qu'est née l'idée du **Mac App Store**.

5. Trouver une application n'a jamais été aussi simple

Le Mac App Store est accessible depuis **fin décembre 2010 – début janvier 2011**, gratuitement pour tous les utilisateurs de **Mac OS X Snow Leopard**, la version actuelle de Mac OS X.



Cet App Store reprend les mêmes éléments que l'App Store iOS :

- accès via le compte iTunes ;
- applications gratuites et payantes ;
- moteur de recherche ;
- commentaires et notes ;
- catégories.

6. La simplicité de l'installation et de la mise à jour



Tout comme sur l'App Store iOS, il vous suffit de sélectionner une application pour la télécharger et l'installer.

Vous serez automatiquement averti des mises à jour disponibles.

Vous ne paierez qu'une fois et téléchargerez quand vous voudrez.

7. La qualité au rendez-vous

Tout comme pour l'App Store iOS, un système de validation sera mis en place par Apple afin de garantir la qualité des applications disponibles.

8. L'accès rapide aux applications

Ce système ne serait pas complet sans un accès aussi simple et rapide aux applications que sur iOS.

Avec le **LaunchPad**, vous aurez automatiquement accès à toutes vos applications, comme sur iPad.



LaunchPad sera intégré à la prochaine version de **Mac OS X Lion (10.7) disponible pour l'été 2011**.

9. Sur le site Apple

- Mac App Store : [Lien 128](#)
- Mac OS X Lion : [Lien 129](#)

Retrouvez l'article de Florent Morin en ligne : [Lien 130](#)

Les derniers tutoriels et articles

Les bases de l'Objective-C

Bonjour à tous, dans ce deuxième tutoriel iPhone, nous allons découvrir les bases du développement sur iPhone/iPad. Cette partie nous permettra de nous familiariser avec l'Objective-C et les outils de développement fournis par Apple (Xcode dans un premier temps).

1. Qu'est-ce qu'Objective-C ?

Objective-C a été créé en 1980 et est une extension du langage C. Il ajoute de nouvelles fonctionnalités au C dont la plus importante est une structure POO (Programmation Orientée Objet ([Lien 131](#))).

1.1. Comment créer une classe ?

La création d'une classe se fait en deux étapes en Objective-C. La première consiste à créer le fichier header ou interface (.h) et la seconde est la création du fichier d'implémentation (.m).

Le fichier header/interface est utilisé pour définir toutes les méthodes et propriétés qui seront utilisées dans votre classe.

Le fichier d'implémentation est le fichier dans lequel vous allez écrire votre code qui fait ce que vous avez décrit dans votre fichier header/interface.

Voici la syntaxe du fichier header :

```
@interface MaClasse : NSObject {
    // vous placez ici toutes vos propriétés
}
// et vos méthodes en dehors des accolades
end
```

Et du fichier d'implémentation :

```
// Vous devez absolument importer votre
fichier .h
#import <MaClasse.h>
@implementation MaClasse
// vous placez ici votre code
end
```

À noter : Xcode vous proposera de créer automatiquement le fichier header lorsque vous créerez le fichier d'implémentation.

1.2. La syntaxe

L'Objective-C possède une syntaxe assez différente des langages de programmation utilisés de nos jours (Java, C++, Ruby, etc.), cependant cela ne le rend pas plus compliqué, bien au contraire. L'Objective-C est fait pour être lu et compris directement par le programmeur. Par exemple pour faire une comparaison entre deux chaînes de caractères en C vous devriez écrire :

```
strcmp('maChaine', 'maChainedeComparaison')
```

En Objective-C, la syntaxe sera différente :

```
[maChaine
isEqualToString:@"maChainedeComparaison"]
```

À noter :

- l'Objective-C étant un « dérivé » du C, vous pouvez coder aussi bien en Objective-C qu'en C, cependant vous devez entourer votre code avec des crochets pour que le compilateur interprète votre code en tant qu'Objective-C et non en C ;
- lorsque vous utilisez une chaîne de caractères Objective-C, vous devez le « signaler » au compilateur en la précédant du signe @.

```
// Chaîne C
myString = "Chaîne C";
// Chaîne Objective-C
myString = @"Chaîne Objective-C";
```

1.3. Les types

Comme nous venons de le préciser un peu plus haut, l'Objective-C est un « dérivé » du langage C, ce qui implique donc que l'Objective-C possède tous les types primitifs de C. Ainsi, vous aurez accès aux types suivants : int, char, short, long, etc.

Cependant, Objective-C arrive avec son lot de nouveaux types. Vous pourrez donc avoir des variables de type NSString pour une chaîne de caractères, NSArray pour stocker vos données dans un tableau, NSDictionary pour vos tableaux key/value. Notez qu'Objective-C introduit un type unique et très intéressant qui est le type « id » qui pourra être utilisé lorsque vous ne connaissez pas le type de votre variable/objet/paramètre.

À noter : sachez que chacun de ces types NS est statique, c'est-à-dire qu'une fois que vous avez attribué une valeur à votre objet, vous ne pourrez la modifier. Cependant, il existe un équivalent dynamique qui vous permettra de changer la valeur de votre objet à volonté en utilisant le mot-clé « Mutable ». Par exemple, le type dynamique pour NSString sera NSMutableString, idem pour NSArray et NSDictionary qui seront NSMutableArray et NSMutableDictionary.

1.4. Les méthodes

Comme tout langage objet, vous avez deux types de méthodes en Objective-C : les méthodes de classe et les méthodes d'instance.

La déclaration d'une méthode en Objective-C se découpe en trois ou quatre parties :

1. vous devez choisir ce que vous déclarez : soit une méthode de classe, soit une méthode d'instance. Les méthodes de classe seront précédées du signe « + », les méthodes d'instance par le signe « - » ;
2. ensuite il vous faudra définir le type que votre méthode retournera entre parenthèses...
3. ...suivi du nom de votre méthode. Essayez d'obéir aux conventions Objective-C en nommant vos méthodes pour qu'on puisse comprendre ce que cette dernière effectuera, et n'oubliez pas le camelCase (première lettre en minuscule (pour les noms des propriétés et méthodes) ou en majuscule (pour les noms de classe) et chaque nouveau mot avec la première lettre en majuscule). Par exemple une méthode qui affiche l'heure sera : « afficherHeure » et non « methode1 » ou encore « AfficherHEURE » ;
4. enfin, viendront se greffer les paramètres, s'il y en a, avec leur type précédé du signe « : ».

Facile non ? Sinon un exemple qui résume les quatre points vous aidera sans doute.

Déclaration d'une méthode dans le fichier header

```
// Déclaration d'une méthode d'instance qui ne
retourne rien et qui prend deux
// paramètres : une propriété de type id et un
nom de type NSString
- (void) changeProperty: (id *) property andName:
(NSString *) name;

// Déclaration d'une méthode de classe sans
paramètre
+ (void) myMethod;
```

Implémentation d'une méthode dans le fichier d'implémentation

```
// Implémentation de notre méthode d'instance qui
ne retourne rien et qui prend
// deux paramètres : une propriété de type id et
un nom de type NSString
- (void) changeProperty: (id *) property andName:
(NSString *) name{
// Insérez ici votre code
}

// Déclaration de notre méthode de classe
+ (void) myMethod{
// Insérez ici votre code
};
```

À noter : si vous souhaitez implémenter vos getters et setters différemment, vous pouvez utiliser les mots-clés « @property » et « @synthesize », cela vous fera gagner du temps et des lignes de code. Vous pourrez par la suite utiliser le « dot-syntax » pour changer ou récupérer la valeur de vos propriétés :

Avant

```
// Fichier .h
- (NSString *) name;
- (void) setName: (NSString *) n;

// Fichier .m
- (NSString *) name{
```

```
return [self name];
}
- (void) setName: (NSString *) n{
// problème de mémoire ici mais nous verrons
dans la suite
name = n;
}

//Appel de la propriété
[object name];
```

Après

```
// Fichier .h
@property (nonatomic, retain) NSString* name;

// Fichier .m
@synthesize name;

//Appel de la propriété
object.name;
```

1.5. La gestion de la mémoire

Voici la partie qui peut causer pas mal de maux de tête, je vous demanderais donc d'être très attentif car la gestion de la mémoire joue un rôle très important dans le développement de votre application iPhone. C'est ce qui va d'ailleurs faire la différence entre une application fluide et une application qui ne fait que crasher.

Contrairement au développement sur Mac, ou encore au développement Java, Cocoa Touch ne vient pas avec un Garbage Collector se chargeant de gérer la mémoire automatiquement. En effet, un Garbage Collector ([Lien 132](#)) demande beaucoup de ressources, or nous sommes sur un mobile où il y a peu de ressources disponibles contrairement à un ordinateur de bureau. Apple a donc décidé de ne pas intégrer un tel système afin d'économiser des ressources système.

Lorsque vous aurez une variable à déclarer, il vous faudra d'abord lui allouer un espace dans la mémoire du téléphone. Lorsque vous aurez cet espace qui vous sera alloué, il vous faudra initialiser cet objet si vous souhaitez l'utiliser. Attention, vous devrez toujours utiliser des pointeurs lorsque vous déclarez vos variables qui auront besoin d'un espace mémoire (autres que types primitifs). Ensuite vous pourrez utiliser toutes vos méthodes d'instance pour effectuer vos actions. Lorsque vous aurez fini avec votre objet, il ne vous restera qu'à le « libérer » de la mémoire du téléphone.

Pour récapituler :

- on alloue un espace mémoire à notre variable :

```
NSString * myString = [NSString alloc];
```

- on initialise notre variable afin de pouvoir appeler les méthodes qui nous intéressent :

```
myString = [NSString init];
```

- on appelle une méthode pour changer le texte de notre string par exemple :


```
// Ici on ajoute un @ devant notre chaîne de
caractères pour signaler
// qu'on utilise une chaîne Objective-C et non C.
[myString setText:@"HelloWorld"];
```

- enfin lorsqu'on a fini avec notre objet on le libère :

```
// Le mot-clé release permet de décrémenter le
retainCount
[NSString release];
```

À noter : en général, le release se fera dans la méthode dealloc de votre objet, en effet, chaque objet héritant de la classe mère « NSObject » possède une méthode « dealloc » dans laquelle nous libérerons tous les objets que nous aurons créés précédemment.

Voici à quoi ressemblera votre méthode d'instance « dealloc » :

```
- (void)dealloc {
    [maPropriété release];
    [super dealloc];
}
```

Comme vous avez pu le lire, le mot-clé « release » ne libère pas vraiment notre objet. Il faut savoir que dès qu'on alloue un espace mémoire à un objet, le retainCount de cet objet se voit incrémenté. Ce retainCount supérieur à 1 permet à l'objet de rester en vie. Ainsi, vous avez la possibilité d'incrémenter ce retainCount grâce au mot-clé « retain », ou de le décrémenter grâce au mot-clé « release ».

Dès que le retainCount est égal à 0, le système se charge de « libérer » l'espace mémoire occupé précédemment par l'objet en question.

Vous avez également la possibilité d'utiliser le mot-clé « autorelease » qui effectue la même action que « release » sauf que ceci se produira plus tard dans le temps. Sachez qu'en permanence, lorsque votre programme est lancé, une boucle tourne en tâche de fond qui s'occupe par exemple de récupérer les événements extérieurs tels qu'un appui sur l'écran de l'iPhone, ou encore d'effectuer l'action autorelease sur un objet lors du prochain passage de la boucle. On parle ici de millisecondes.

1.6. À retenir

```
// alloue un espace mémoire + initialise l'objet
NSString * myObject = [[NSString alloc] init] //
retainCount = 1

// augmente de retainCount de + 1
[myObject retain]; //retainCount = 2

// on diminue le retainCount de -1 et encore de
-1
// ce qui a comme effet de libérer notre objet
[[myObject release] release]; // retainCount = 0
```

À noter : il ne faut jamais appeler la méthode release sur un objet dont vous n'avez pas le contrôle. En effet, vous devez appeler la méthode release seulement sur les objets

qui vous appartiennent, c'est-à-dire ceux que vous avez créés.

1.7. Récapitulatif

Pfiou, on en voit presque le bout. Avant de passer au développement de notre toute première application iPhone dans le prochain tuto, nous allons récapituler tout ce que nous venons de voir à travers une classe Voiture qui aura deux attributs (marque et couleur) et une méthode d'initialisation qui définira la marque et la couleur de la voiture.

Fichier : header/interface // Voiture.h

```
#import <Cocoa/Cocoa.h>
@interface Voiture : NSObject {
    NSString * couleur;
    NSString * marque;
}
// Getters et setters
@property (nonatomic, retain) NSString* couleur;
@property (nonatomic, retain) NSString* marque;
end
```

Fichier implementation // Voiture.m

```
#import <Voiture.h>
@implementation Voiture

- (id) initWithColor: NSString* color andMarque:
NSString* model
{
    // Ici on instancie notre classe grâce à notre
    classe NSObject
    // et on vérifie si l'instanciation s'est bien
    déroulée
    if (self = [super init])
    {
        self.couleur = color;
        self.marque = model;
    }
}

@synthesize couleur;
@synthesize marque;

- (void) dealloc
{
    [couleur release];
    [marque release];
    [super dealloc];
}
end
```

Utilisation de notre classe :

```
//retainCount = 1
Voiture * mercedes = [[Voiture alloc]
initWithColor:@"gris" andMarque:@"Mercedes"];
// ....
[mercedes release]; // reteainCount = 0 ce qui
appelle la méthode dealloc
```

Voilà c'est terminé pour ce deuxième tutoriel iPhone. Rendez-vous très vite pour le prochain. Il vous permettra de réaliser votre première application qui sera un classique mais efficace Hello World.

Retrouvez l'article d'Axon en ligne : [Lien 133](#)

Développer votre webapp mobile avec Wink

Wink est un framework JavaScript permettant de développer des applications Web sur mobile. C'est un outil particulièrement adapté aux navigateurs WebKit que l'on retrouve aujourd'hui sur l'iPhone, sur Android ou encore les terminaux BlackBerry.

1. Description

Wink est un framework JavaScript mobile Open Source et un projet de la fondation Dojo. Il cible avant tout les navigateurs WebKit (que l'on retrouve sur la majorité des smartphones et tablettes du moment) et annonce sa compatibilité avec iOS (iPod touch, iPhone, iPad), Android ou encore BlackBerry. Il a été pensé et construit pour s'adapter aux contraintes et aux spécificités des environnements Web mobile : c'est un framework qui reste ultra-léger (6 ko une fois minimisé et gzippé) et qui fournit, par exemple, toute une couche de gestion des événements "touch" et "gesture". Bien entendu, on y retrouve les fonctions HTML5 comme la géolocalisation ou le stockage local. Toutes les animations (2D et 3D) sont quant à elles réalisées en CSS3.

Wink présente un grand nombre d'éléments graphiques innovants (nuages de tags 3D en HTML, mises en pages originales, composants multitouch...) et se démarque ainsi de ses concurrents directs comme Sencha ou jQuery mobile.

Enfin, Wink se veut être un framework simple d'utilisation, c'est ce que nous allons voir au travers des différents exemples présentés dans cet article.

Ci-dessous, un exemple de composant innovant que l'on retrouvera dans le toolkit : [Lien 134](#)

2. Présentation des packages

Wink a été divisé en huit packages

Wink Core, qui constituent la base du framework, offrent toutes les fonctions de base dont un développeur a besoin. Des requêtes XHR à la manipulation du DOM, en passant par la publication d'événements ou la gestion des transformations CSS. C'est aussi dans ce package que l'on retrouve tout ce qui concerne la gestion des événements de toucher.

Wink "ui" rassemble tous les composants graphiques. Certains sont familiers, comme le "date picker" ou le "carrousel" alors que d'autres sont plus inattendus comme le nuage de tags HTML en 3D ou le composant "flippage" permettant de naviguer à la manière d'un e-book. C'est dans ce package que l'on retrouve aussi les layouts les plus populaires : sliding panels, scrollers... De quoi donner à sa webapp un Look and Feel d'application native.

Quelques vidéos des composants en action

- Carrousel : [Lien 135](#)
- Nuage de tags 3D : [Lien 136](#)
- Scroller : [Lien 137](#)

C'est dans Wink "ux" que l'on retrouve tous les composants d'interaction. Bien sûr, les éléments "touch" et "multitouch" mais aussi des briques comme le gestionnaire de Drag and Drop ou encore l'adaptation du moteur de reconnaissance de forme "\$1". Des utilitaires permettant de facilement gérer les changements d'orientation du terminal, ou encore l'historique de navigation du browser sont également présents.

Quelques vidéos des composants en action

- Drag and Drop : [Lien 138](#)
- Reconnaissance de forme : [Lien 139](#)

Wink "net" inclut un certain nombre de "loaders". Le "JsLoader" peut être utilisé pour charger des ressources JavaScript externes à la volée, ce qui devient vite utile dans un environnement mobile dans lequel on voudra éviter de charger toutes les ressources au démarrage. Ce composant est parfaitement adapté pour le JSONP. Le préchargement d'images et de CSS fait aussi partie de ce package. Comme on le verra plus amplement par la suite, l'instanciation des composants reste simple et suit toujours le même pattern.

Voilà un premier exemple d'instanciation du JsLoader utilisé pour effectuer une requête JSONP

```
jsLoader = new wink.net.JsLoader();  
jsLoader.load('http://server2.example.com/getjson  
?jsonp=parseResponse');
```

Wink "fx" regroupe toutes les bibliothèques CSS permettant de gérer les transformations 2D, 3D ainsi que les animations. Wink "api" fournit des couches d'abstraction aux API HTML5 comme la géolocalisation ou le stockage local et le tout nouveau Wink "mm" (multimédia) contient désormais des players audio et vidéo.

Un tout nouveau composant "Easy Caching" vient compléter le tout. Le but : répliquer le fonctionnement du cache grâce au stockage local et ainsi améliorer jusqu'à dix fois les temps de chargement des WebApps tout en s'affranchissant des problèmes de compatibilité entre les différents OS mobiles.

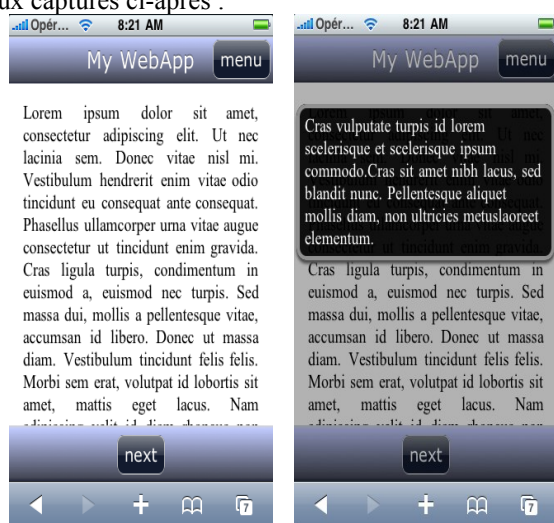
3. Téléchargement et utilisation

L'intégralité du framework est disponible sur winktoolkit.org ([Lien 140](#)). Deux choix s'offrent alors à vous. Soit télécharger la version "code source", soit la version optimisée. Le plus simple est de télécharger cette dernière car elle comprend le core déjà minimisé. Dans cet article, nous utiliserons la version 1.2 optimisée que vous pourrez retrouver ici : [Lien 141](#).

4. Ma première WebApp

Nous allons maintenant développer notre première WebApp en se basant en partie sur les composants Wink. Le but est d'obtenir une WebApp "classique" contenant un header et un footer fixes, une zone de contenu intermédiaire dans laquelle l'utilisateur pourra scroller. Enfin, un bouton dans le header permettra l'ouverture d'un menu en overlay dans la page.

Au final, nous devrions obtenir un résultat équivalent aux deux captures ci-après :



Dans un premier temps, nous allons créer un répertoire test qui contiendra notre WebApp. Le fichier principal de notre application sera index.html (situé dans le répertoire test). Pour simplifier, le répertoire wink est recopié directement dans le répertoire test, afin que notre page HTML accède facilement aux ressources.

4.1. Première étape : positionnement des metas

Nous allons définir deux directives meta dans cette page. La première spécifie que sur iOS, la WebApp va pouvoir s'afficher en mode "WebApp" (plus d'informations sur le mode "WebApp" : [Lien 141](#)). La seconde va permettre d'adapter la page HTML à la taille de l'écran.

index.html : metas

```
<meta name="apple-mobile-web-app-capable"
content="yes">
<meta name="viewport" content="width=device-
width; initial-scale=1.0; maximum-scale=1.0;
user-scalable=no;" />
```

4.2. Deuxième étape : création des header et footer

Pour accéder aux fonctionnalités du toolkit, il va tout d'abord falloir rajouter un lien vers le core de Wink (fichier wink.min.js) dans la section "head" de notre WebApp.

index.html : layout

```
<script type="text/javascript"
src="../../js/wink.min.js"></script>
```

Nous allons maintenant créer le layout de la page. À savoir : un header, une zone de contenu et un footer.

index.html : layout

```
<body>
    <div id="header">My WebApp</div>
    <div id="content">
        Lorem ipsum dolor sit amet...
    </div>
    <div id="footer">
    </div>
</body>
```

Enfin, appliquons les styles sur les différents éléments HTML de la page.

index.html : styles

```
#header, #footer {
    position: relative;
    z-index: 0;
    width: 100%;

    text-align: center;
    font-size: 1.4em;
    line-height: 2em;
    font-family: Verdana, Helvetica, sans-
serif;
}

#header {
    background-color: #444755;
    background-image: -webkit-
gradient(linear, 0 0, 0 100%, color-stop(0,
#d4d4d4),
                                color-
stop(0.05, #c1cbff), color-stop(0.95, #444755),
color-stop(1, #23273b));
    color: #fff;
    text-shadow: 0 -0.05em 0 rgba(0, 0, 0,
0.5);
}

#footer {
    background-color: #444755;
    background-image: -webkit-
gradient(linear, 0 0, 0 100%, color-stop(0,
#d4d4d4),
                                color-
stop(0.05, #c1cbff), color-stop(0.95, #444755),
color-stop(1, #23273b));
    color: #fff;
}

#content {
    position: relative;
    width: 100%;
    min-height: 300px;
    -webkit-box-sizing: border-box;
    padding: 1em;
    font-size: 1.1em;
    text-align: justify;
    background-color: #fff;
    color: #000;
}
```

4.3. Troisième étape : ajout du menu

Nous allons maintenant devoir nous servir du composant Popup de Wink (situé dans le répertoire ui/xy/popup). Pour cela, il va donc falloir rajouter dans la section "head" de la WebApp un lien vers le fichier JavaScript du composant Popup et un lien vers la feuille de style qui lui est associée.

index.html : intégration du composant Popup

```
<link rel="stylesheet"
href="../../../wink/ui/xy/popup/css/styles.css"
type="text/css">
<script type="text/javascript"
src="../../../wink/ui/xy/popup/js/popup.js"></script>
```

Une fois cette étape effectuée, il va falloir rajouter un bouton dans le header. C'est sur ce bouton qu'il faudra cliquer pour voir le menu apparaître. Nous allons donc modifier la structure HTML. Vous remarquerez que nous avons rajouté dans la balise body un handler sur l'événement load qui va permettre, une fois que la page sera chargée, de déclencher l'appel à la fonction "webapp.init" qui sera détaillée ci-après.

index.html

```
<body onload="setTimeout(webapp.init, 1);">
  <div id="header">
    <div class="title"><span>My
WebApp</span></div>
    <div id="menuBtn"
class="button"><span>menu</span></div>
  </div>
  <div id="content">
    Lorem ipsum dolor sit amet...
  </div>
  <div id="footer"> </div>
</body>
```

Intégrons le style du bouton dans la balise style précédemment créée.

index.html : styles

```
.title {
  display: inline-block;
}

.button {
  color: #fff;
  display: inline-block;
  padding: 0em 0.5em;
  height: 2em;
  line-height: 2em;
  font-size: 0.8em;
  background-image: -webkit-
gradient(linear, left top, left bottom,
from(#7d828c),
                                color-
stop(0.5, #303749), color-stop(0.5, #121a2e),
to(#121a2e));
  border: solid 0.1em rgba(240, 240, 240,
0.75);
  -webkit-user-select: none;
  -webkit-border-radius: 0.5em;
}

#header .button {
  position: absolute;
  top: 0.2em;
```

```
right: 0.2em;
}
```

Reste à instancier et à initialiser le composant Popup, et à gérer les clics sur notre bouton.

Instanciation du composant Popup et gestion des clics sur le bouton contenu dans le header

```
var webapp =
{
  popup: null,

  init: function()
  {
    window.scrollTo(0,0);
    webapp.initPopup();
  },

  initPopup: function()
  {
    webapp.popup = new wink.ui.xy.Popup()
    document.body.appendChild(webapp.popup.getDom
Node());

    $('menuBtn').onclick = function() {
      var menu = "";
      menu += "<div class='menu'>";
      menu += "Cras vulputate turpis id lorem
scelerisque et scelerisque ipsum commodo.";
      menu += "Cras sit amet nibh lacus, sed
blandit nunc. Pellentesque aliquet mollis diam,
non ultricies metus";
      menu += "laoreet elementum.";
      menu += "</div>";

      webapp.popup.popup({
        content: menu,
        top: ($('header').offsetHeight + 20) + "px"
      });
    }
  }
};
```

4.4. Quatrième étape : les sliding panels

Si vous voulez donner à votre site Web des airs d'application native, vous allez sans doute vouloir que la navigation entre les différentes pages de votre WebApp se fasse en "slidant". Pour cela, Wink offre un composant adapté qui permet de choisir parmi trois types de slides différents ("normal", "cover" et "reveal") tout en spécifiant la durée de l'effet.

Nous allons maintenant devoir utiliser le composant SlidingPanels de Wink (situé dans le répertoire ui/layout/slidingpanels). Encore une fois, il va falloir rajouter dans la section "head" de la WebApp un lien vers le fichier JavaScript du composant ainsi qu'un lien vers la feuille de style qui lui est associée.

index.html : intégration du composant SlidingPanels

```
<link rel="stylesheet"
href="../../../wink/ui/layout/slidingpanels/css/s
tyles.css" type="text/css">
<script type="text/javascript"
src="../../../wink/ui/layout/slidingpanels/js/sli
dingpanels.js"></script>
```

Le corps du document HTML doit aussi être modifié en fonction pour intégrer la notion de page. On pensera à bien séparer les pages identifiées par une 'div' (page1, page2) de leur contenu via une autre div (content1, content2) ; ceci afin de pouvoir associer le style souhaité au contenu sans modifier le style des pages qui est propre au fonctionnement du composant. Il nous faut aussi des boutons pour pouvoir naviguer à travers les pages. Nous allons intégrer cette fonctionnalité dans le footer de la WebApp.

index.html

```
<body onload="setTimeout(webapp.init, 1);">
  <div id="header">
    <div class="title"><span>My
WebApp</span></div>
    <div id="menuBtn"
class="button"><span>menu</span></div>
  </div>
  <div id="content">
    <div id="page1">
      <div id="content1" class="pageContent">
        Lorem ipsum dolor sit amet ...
      </div>
    </div>
    <div id="page2">
      <div id="content2" class="pageContent">
        Aliquam vehicula scelerisque ...
      </div>
    </div>
  </div>
  <div id="footer">
    <div id="previousBtn" class="button
hidden"><span>previous</span></div>
    <div id="nextBtn"
class="button"><span>next</span></div>
  </div>
</body>
```

Intégrons ensuite les styles de nos pages

index.html : styles

```
.pageContent {
  min-height: 300px;

  padding: 1em;
  font-size: 1.1em;
  text-align: justify;

  -webkit-box-sizing: border-box;
}

#content1 {
  background-color: #fff;
  color: #000;
}

#content2 {
  background-color: #000;
  color: #fff;
}
```

Reste à déclencher la création du composant SlidingPanels, en JavaScript, au démarrage de l'application. Reste aussi à gérer les boutons "suivant" et "précédent" qui permettront de naviguer dans la webapp. Pour cela nous allons rajouter plusieurs méthodes et modifier la fonction "init".

Instanciation du composant SlidingPanels et gestion des clics sur les boutons suivant/précédent

```
var webapp =
{
  popup: null,

  init: function()
  {
    window.scrollTo(0,0);
    webapp.initSize();
    webapp.initPopup();
    webapp.initPanels();

    $('nextBtn').onclick = webapp.next;
    $('previousBtn').onclick = webapp.previous;
  },

  initPopup: function()
  {
    webapp.popup = new wink.ui.xy.Popup()
    document.body.appendChild(webapp.popup.getDomNode());

    $('menuBtn').onclick = function() {
      var menu = "";
      menu += "<div class='menu'>";
      menu += "Cras vulputate turpis id lorem scelerisque et scelerisque ipsum commodo.";
      menu += "Cras sit amet nibh lacus, sed blandit nunc. Pellentesque aliquet mollis diam, non ultricies metus";
      menu += "laoreet elementum.";
      menu += "</div>";

      webapp.popup.popup({
        content: menu,
        top: $('header').offsetHeight + 20 + "px"
      });
    };
  },

  initPanels: function()
  {
    webapp.slidingPanels = new
wink.ui.layout.SlidingPanels(
    {
      'duration': 500,
      'pages': [ 'page1', 'page2' ]
    });

    $('content').
appendChild(webapp.slidingPanels.getDomNode());
  },

  next: function()
  {
    wink.addClass($('nextBtn'), "hidden");
    wink.removeClass($('previousBtn'), "hidden");

    webapp.slidingPanels.slideTo('page2');
  },

  previous: function()
  {
    wink.removeClass($('nextBtn'), "hidden");
    wink.addClass($('previousBtn'), "hidden");

    webapp.slidingPanels.slideBack();
  }
}
```

```

},

initSize: function()
{
    $('content').style.height =
    pagel.offsetHeight + "px";
}
};

```

4.5. Cinquième étape : le scroller (header et footer fixes)

Pour terminer la WebApp et lui permettre d'avoir un comportement proche de celui d'une application native, nous allons utiliser le composant Scroller. Il va nous permettre de garder les header et footer fixes tout en pouvant naviguer dans nos deux pages.

index.html : intégration du composant Slider

```

<script type="text/javascript"
src="../../wink/ux/window/js/window.js"></script>
<script type="text/javascript"
src="../../wink/ux/movementtracker/js/movement
tracker.js"></script>
<script type="text/javascript"
src="../../wink/ux/inertia/js/inertia.js"></script>
<script type="text/javascript"
src="../../wink/ui/layout/scroller/js/scroller
.js"></script>

```

Pour utiliser ce composant, il va simplement falloir rajouter son instanciation à l'initialisation de la WebApp.

Instanciation du composant Scroller

```

var webapp =
{
    popup: null,

    init: function()
    {
        window.scrollTo(0,0);
        webapp.initSize();
        webapp.initScroller();
        webapp.initPopup();
        webapp.initPanels();

        $('nextBtn').onclick = webapp.next;
        $('previousBtn').onclick = webapp.previous;
    },

    initPopup: function()
    {
        webapp.popup = new wink.ui.xy.Popup()
        document.body.appendChild(webapp.popup.getDomNode());

        $('menuBtn').onclick = function() {
            var menu = "";
            menu += "<div class='menu'>";
            menu += "Cras vulputate turpis id lorem
scelerisque et scelerisque ipsum commodo.";
            menu += "Cras sit amet nibh lacus, sed
blandit nunc. Pellentesque aliquet mollis diam,
non ultricies metus";
            menu += "laoreet elementum.";
            menu += "</div>";

            webapp.popup.popup({

```

```

        content: menu,
        top: ($('#header').offsetHeight + 20) + "px"
    });
    },

    initPanels: function()
    {
        webapp.slidingPanels = new
wink.ui.layout.SlidingPanels(
        {
            'duration': 500,
            'pages': [ 'page1', 'page2' ]
        });

        $('content').
appendChild(webapp.slidingPanels.getDomNode());
    },

    initScroller: function()
    {
        webapp.scroller1 = new
wink.ui.layout.Scroller(
        {
            target: "content1",
            direction: "y"
        });

        webapp.scroller2 = new
wink.ui.layout.Scroller(
        {
            target: "content2",
            direction: "y",
            scrollbars: {
                backgroundColor: 'rgba(255, 255, 255,
0.8)',
                borderColor: 'rgba(255, 255, 255, 0.9)'
            }
        });
    },

    next: function()
    {
        wink.addClass($('nextBtn'), "hidden");
        wink.removeClass($('previousBtn'), "hidden");

        webapp.slidingPanels.slideTo('page2');
    },

    previous: function()
    {
        wink.removeClass($('nextBtn'), "hidden");
        wink.addClass($('previousBtn'), "hidden");

        webapp.slidingPanels.slideBack();
    },

    initSize: function()
    {
        $('content').style.height =
        pagel.offsetHeight + "px";
    }
};

```

5. Pour aller un peu plus loin

D'autres exemples de code sont à retrouver dans la section "tutorials" du site de Wink ([Lien 143](#)), ainsi que des conseils sur le développement Web mobile.

Un forum de discussion est aussi là pour aider les développeurs à résoudre les problèmes rencontrés avec l'outil.

Enfin, si vous êtes intéressé pour contribuer au projet Wink, n'hésitez pas à contacter ses créateurs ([Lien 144](#)).

Retrouvez l'article de Jérôme Giraud en ligne : [Lien 145](#)

Plus loin avec MS Graph - Découvrez d'autres propriétés de MS Graph

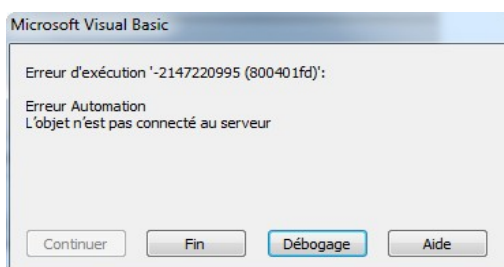
Ce cours est le deuxième consacré à la programmation de MS Graph. La première partie nous a donné un bref aperçu du composant MS Graph. Ce cours va être l'occasion d'aller plus loin avec ce composant essentiel de Microsoft.

1. Introduction

La première partie de cette série d'articles nous a permis de faire une rapide prise en main du composant **MS Graph**. Dans cette partie nous allons nous intéresser à la mise en forme de notre graphique.

1.1. Avertissement

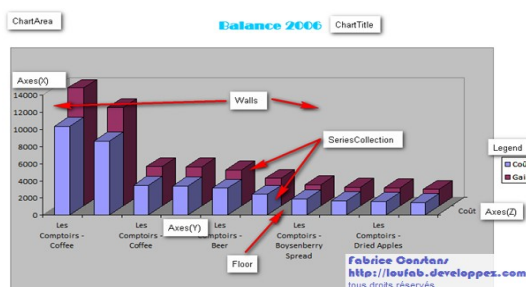
L'utilisation d'un objet **MS Graph** nécessite qu'il soit toujours actif. Ce cours étant composé d'exercices distincts, il peut arriver que le code soit réinitialisé lors des modifications successives du code. Dans ce cas le message suivant apparaîtra.



Relancez le formulaire pour que les variables objets soient réinitialisées et que le code se remette à fonctionner.

2. Les objets

Pour aller plus loin dans l'exploration de **MS Graph** il est important de détailler les objets qu'il contient. L'image suivante indique les principaux objets à manipuler.



Notez que certains objets ne sont disponibles que pour des graphiques 3D.

3. Police, taille et style

Pour mettre en forme le texte nous disposons de l'objet **Font** qui comporte plusieurs propriétés. Il est disponible dans chaque objet comportant du texte (**ChartTitle**, **Legend**). Pour notre exemple nous utiliserons l'objet **ChartArea** qui désigne la zone d'affichage du graphique.

Il contient tous les autres objets du graphique qui hériteront de fait de la configuration.

Commencez par déclarer l'objet dans l'en-tête du module.

```
ChartArea
Dim vlArea As Graph.ChartArea
```

Puis dans la méthode « Initialiser ».

```
ChartArea
Set vlArea = vlChart.ChartArea
'la zone du graphique
```

Puis la procédure qui nous permettra de régler la police.

```
Font
Public Sub pFont(vFontName As String, vFontStyle
As String, vFontSize As Integer)
    vlArea.Font.Name = vFontName
    vlArea.Font.FontStyle = vFontStyle
    vlArea.Font.Size = vFontSize
End Sub
```

L'appel se fait simplement à partir de notre bouton « Modifier ».

```
vGraph.pFont "Calibri", "Normal", 10
```

La totalité des objets de type texte a hérité des nouveaux réglages. Ceci ne nous empêche pas de définir un autre réglage pour un objet contenu dans le graphique.

Repérez la procédure « AfficherTitrePrincipal » de la classe et ajoutez le code ci-dessous.

```
Font
vlChart.ChartTitle.Text = vlTitrePrincipal
With vlChart.ChartTitle.Font
    .Name = "Broadway"
    .Bold = True
    .Size = 14
    .Color = RGB(0, 128, 255)
End With
```

Bold permet de mettre la police en gras et **Color** gère la couleur. Cette dernière accepte une valeur de type **Long**.

Consultez l'aide et la liste de l'auto complétion pour les autres propriétés disponibles.

4. Les séries de données

SeriesCollection est une collection qui regroupe les séries de données. Chaque série de données est organisée suivant le **plotby** (voir le cours précédent).

Actuellement le graphique comprend la série intitulée **Coût** et celle intitulée **Gain**. Nous allons faire quelques manipulations de base pour nous familiariser avec cet objet.

Avant de créer la méthode vous devez inscrire dans les références de VBE la bibliothèque **Microsoft Office xx.x Object Library**. En effet nous allons devoir faire appel à des constantes contenues dans cette bibliothèque.

Dans la classe créez la méthode ci-dessous.

```
SeriesCollection
Public Sub ColoreGraphique()
Dim gSeries As Graph.SeriesCollection
Dim i As Integer
Set gSeries = vlChart.SeriesCollection
For i = 1 To gSeries.Count
With gSeries(i)
.Border.Weight = xlThin
.Border.LineStyle = xlAutomatic
.Fill.TwoColorGradient
msoGradientHorizontal, 1
.Fill.ForeColor.SchemeColor = 2 + i ' une
couleur > à 3
.Fill.BackColor.SchemeColor = 2
End With
Next
Set gSeries = Nothing
End Sub
```

Pas de difficulté pour la création de l'objet « gSeries » qui va nous servir à parcourir les séries de données. Vous pouvez utiliser une boucle **For Each**, cependant dans notre cas nous avons besoin d'un compteur c'est pour cette raison que j'ai opté pour une boucle de type **For** classique.

Info ! Les collections du modèle **MS Graph** démarrent à 1 et non à 0.

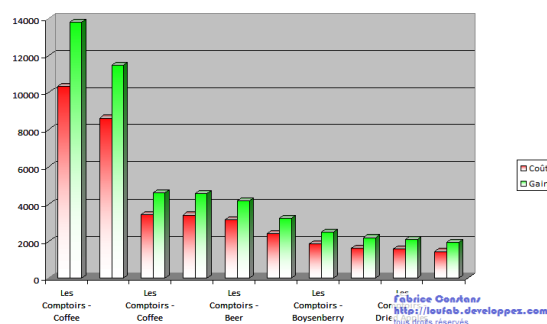
Comme son nom l'indique elle colore les barres de chaque série suivant la couleur définie dans la table des couleurs de **MS Graph**.

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49
50	51	52	53	54	55	56

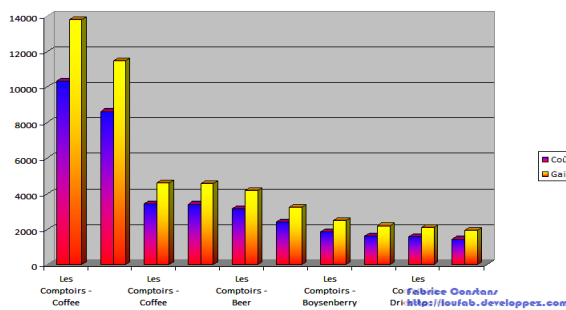
Border gère la bordure des barres. Deux propriétés permettent de régler le trait fin et le style de ligne en automatique.

Fill gère le remplissage de chaque barre de la série.

TwoColorGradient met un effet de dégradé composé de deux couleurs. La couleur principale désignée par **ForeColor** est définie suivant une couleur supérieure à l'index numéro 3, donc ni noir ni blanc. La seconde, **BackColor** est définie en blanc (index numéro 2). Vous pouvez utiliser toutes les combinaisons que vous souhaitez.



En changeant les paramètres **SchemeColor** on peut obtenir de jolis dégradés.



Pour finir avec la gestion des couleurs la propriété **TwoColorGradient** accepte d'autres variations que le dégradé horizontal. Consultez l'aide **Ms Graph** pour en savoir plus.

Les constantes **msoGradient** sont contenues dans la bibliothèque que nous venons de déclarer.

5. Les étiquettes de données

Les **DataLabels** représentent les étiquettes de données qui peuvent être affichées au-dessus des barres. Généralement elles affichent la valeur, mais il faut savoir qu'elles peuvent également afficher la catégorie, la série, le pourcentage pour les types de graphiques qui acceptent cette donnée comme les camemberts par exemple.

L'exemple suivant affiche toutes les étiquettes de toutes les séries.

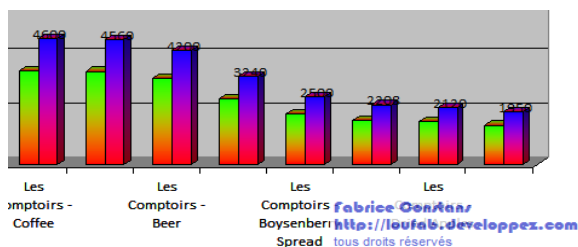
```
SeriesCollection
Public Sub AfficheLabels()
Dim gSeries As Graph.SeriesCollection
Dim gSerie As Graph.Series
Set gSeries = vlChart.SeriesCollection
For Each gSerie In gSeries
gSerie.HasDataLabels = True
Next
Set gSeries = Nothing
End Sub
```

La propriété **HasDataLabels** indique si l'affichage des

étiquettes est activé ou non. Rien ne nous empêche de les afficher pour une seule série.

HasDataLabels

```
vlChart.SeriesCollection.Item(1).HasDataLabels = True
vlChart.SeriesCollection.Item(2).HasDataLabels = False
```



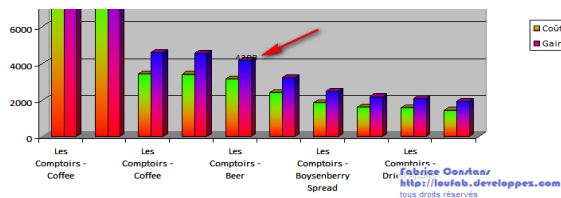
5.1. Points

Collection sous-ensemble de la série, **Points** permet de faire référence à une position de la série. En l'occurrence une barre d'une série. Un exemple concret avec ce code qui permet de mettre une étiquette sur la sixième barre de la première série.

HasDataLabel

```
With myChart.SeriesCollection(1).Points(6)
    .HasDataLabel = True
End With
```

Attention ! Les étiquettes ne doivent pas être affichées pour la première série.



On peut également contrôler le style d'affichage de l'étiquette.

DataLabel

```
Public Sub gGerePoint()
    Dim gPoints As Graph.Points
    Set gPoints =
vlChart.SeriesCollection(1).Points
    With gPoints.Item(5)
        .HasDataLabel = True
        .DataLabel.Font.ColorIndex = 2
        .DataLabel.Fill.ForeColor.SchemeColor = 3
    End With
    Set gPoints = Nothing
End Sub
```

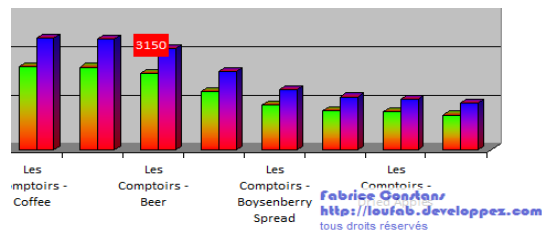
La couleur de la police est le blanc et le fond de l'étiquette est rouge.

Info ! Lorsque vous réglez la couleur de la police en blanc son fond est automatiquement mis en noir.

L'étiquette est remontée de 20 points (1 inch = 72 points).

DataLabel

```
.DataLabel.Top = .DataLabel.Top - 20
```



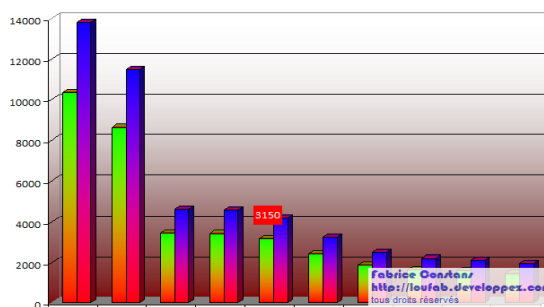
6. Fond et plancher

MS Graph est un produit complet. Il est donc normal que l'on puisse tout contrôler. Les murs, **Walls** et **Floor**, disposent des mêmes propriétés de mise en couleur. L'exemple suivant montre comment donner de la couleur à l'arrière-plan du graphique.

Walls

```
Public Sub gGereMurs()
    Dim gWalls As Walls
    Set gWalls = vlChart.Walls
    With gWalls.Fill
        .ForeColor.SchemeColor = 9
        .BackColor.SchemeColor = 2
        .TwoColorGradient
        Style:=msoGradientHorizontal, variant:=2
    End With
    Set gWalls = Nothing
End Sub
```

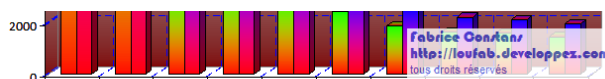
Deux couleurs sont assemblées avec un dégradé horizontal.



Pour le fond la méthode est la même pour l'objet **Floor**. Le code suivant permet de colorer le plancher en blanc (index numéro 2).

Floor

```
Public Sub gGerePlanche()
    Dim gFloor As Floor
    Set gFloor = vlChart.Floor
    gFloor.Interior.ColorIndex = 2
    Set gFloor = Nothing
End Sub
```



7. Grilles

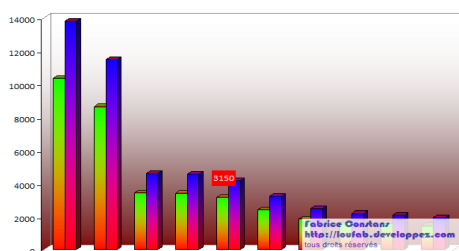
L'aspect des grilles est également modifiable. Elles appartiennent aux axes **xlValue** et **xlCategory** toujours conditionnés par le **PlotBy**.

Il existe deux types de grilles, **MajorGridlines** et **MinorGridlines**, qui se paramètrent toutes les deux sur les deux axes.

Vous pouvez les rendre totalement invisibles en paramétrant leurs propriétés respectives **HasMajorGridlines** et **HasMinorGridlines** à **False**.

HasMajorGridlines

```
Public Sub gGridLines()  
    vlChart.Axes(xlValue).HasMajorGridlines =  
False  
    vlChart.Axes(xlCategory).HasMajorGridlines =  
False  
End Sub
```

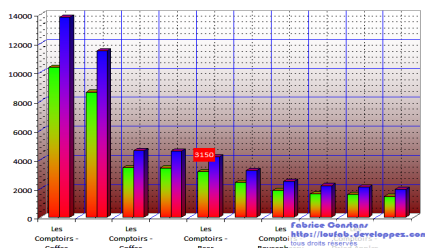


Le code suivant exploite la totalité des **Gridlines**.

GridLines

```
With vlChart.Axes(xlValue)  
    .HasMajorGridlines = True  
    .MajorGridlines.Border.Color = RGB(0, 0, 255)  
    .MajorGridlines.Border.LineStyle =  
xlContinuous  
    .HasMinorGridlines = True  
    .MinorGridlines.Border.LineStyle = xlDot  
End With  
With vlChart.Axes(xlCategory)  
    .HasMajorGridlines = True  
    .MajorGridlines.Border.Color = RGB(0, 0, 255)  
    .MajorGridlines.Border.LineStyle =  
xlContinuous  
    .HasMinorGridlines = True  
    .MinorGridlines.Border.LineStyle = xlDot  
End With
```

Border représente la ligne affichée. Sa propriété **Color** permet d'en définir la couleur comme déjà vu précédemment. **LineStyle** dont les paramètres sont les constantes de type **xlLineStyle** définit le style de ligne en pointillé, tiret.



L'exemple précédent est bien trop chargé pour permettre

une lecture efficace du graphique.

8. Gestion de la 3D

Le graphique de type 3D n'en a pas que le nom, en effet comme dans Excel on peut agir sur les trois axes.

8.1. Précaution

Avant de modifier ces propriétés, il est conseillé de noter les valeurs d'origine. En effet une fois modifiées celles-ci seront perdues.

8.2. Perspective

D'une valeur de 0 à 100. Elle n'est disponible que si la propriété **RightAngleAxes** est à Faux. Ceci est logique puisque **RightAngleAxes** indique que les axes Catégorie et Valeur sont à angle droit.

A 0 il n'y a aucune perspective. Le plancher, **Floor**, apparaît comme une ligne horizontale.

8.3. Elevation

Comprise entre -90 et +90. Elle indique la position de l'oeil dans le plan vertical.

- -90 : le graphique est vu de dessous.
- 0 : vous êtes en face.
- +90 : vous vous tenez au dessus.

8.4. Rotation

Comprise entre 0 et 360°. Le graphique tourne autour d'un axe vertical.

- 0° ou 360° : vous êtes face à lui.
- 90° : le mur de l'axe des valeurs, **xlValue**, est à l'arrière-plan.
- 190° : vous le regardez de dos. Dans cette position le fond, **Walls**, est au premier plan.
- 270° : le mur de l'axe des valeurs, **xlValue**, est au premier plan.

8.5. Rotation, Elevation, Perspective

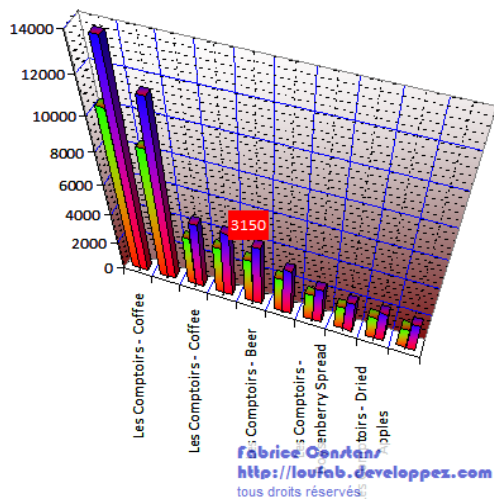
L'exemple suivant met en lumière les propriétés agissant sur la vue 3D du graphique.

3D

```
Public Sub g3dRotate(pPerspective As Integer,  
pElevation As Integer, pRotation As Integer)  
With vlChart  
    .RightAngleAxes = False  
    .Perspective = pPerspective  
    .Elevation = pElevation  
    .Rotation = pRotation  
End With  
End Sub
```

Un petit test avec les valeurs suivantes.

```
vGraph.g3dRotate 50, 50, 20
```

J'espère que vous n'avez pas le vertige.

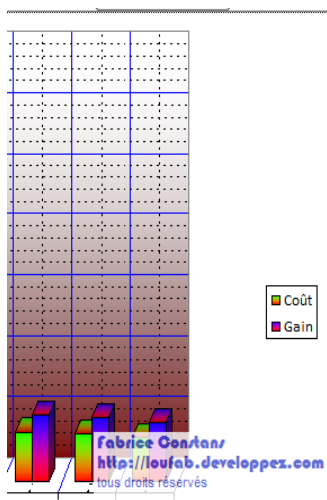
Pour revenir à une représentation classique réglez la propriété **RightAngleAxes** à Vrai et réattribuez les valeurs que vous avez dû conserver conformément aux conseils donnés en début de chapitre.

9. Légende

La légende, **Legend**, est un objet complexe au même titre que l'objet **ChartArea**. Il contient la collection de lignes, **LegendEntries**, représentant chaque Série.

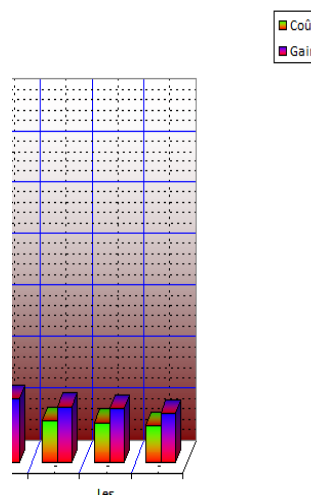
Comme pour **ChartTitle** la propriété **HasTitle** doit être paramétrée à Vrai.

Dans l'exemple suivant nous allons déplacer la légende à l'aide de la propriété **Position**.



HasLegend

```
Public Sub gManipulerLégende()
    Dim vlegend As Legend
    Set vlegend = vlChart.Legend
    vlChart.HasLegend = True
    vlegend.Position = xlLegendPositionCorner
    Set vlegend = Nothing
End Sub
```



Les positions peuvent être définies grâce aux membres de **xlLegendPosition**.

La propriété **Shadow** permet de mettre un ombrage sur le cadre.

Shadow

```
vlChart.HasLegend = True
    vlegend.Position = xlLegendPositionCorner
    vlegend.Shadow = True
Set vlegend = Nothing
```



Comme la plupart des objets, les caractères (**Font**), l'aspect intérieur (**Fill**) et le cadre (**Border**) peuvent être modifiés.

9.1. Les entrées

Chaque ligne de la légende peut être manipulée indépendamment. Dans l'exemple suivant nous mettons la taille de la police de la deuxième ligne à `taille+2`.

LegendEntries

```
vlChart.Legend.LegendEntries(2).Font.Size =
vlChart.Legend.LegendEntries(2).Font.Size + 2
```

Comme avec les autres collections d'objets l'indice commence à 1.

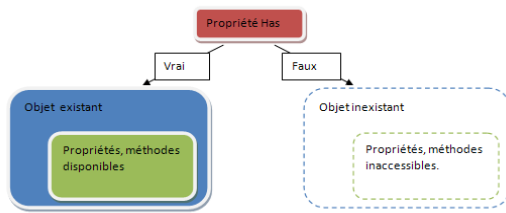
La méthode irréversible **Delete** permet de supprimer la ligne choisie.

10. Particularité : Les propriétés Has

Chaque propriété **Has** (**HasTitle**, **HasLegend**...) permet l'affichage et l'accès à un objet.

Il faut garder à l'esprit que lorsque la propriété est mise à Faux l'objet est détruit.

- Tenter d'y accéder provoque un message d'erreur.
- Les réglages effectués précédemment sont irrémédiablement perdus.



11. Conclusion

Voici ce nouveau chapitre sur **MS Graph** achevé et comme vous avez pu le voir les possibilités de manipulation sont nombreuses et les objets sont plus ou moins complexes.

N'oubliez pas de consulter la documentation de l'objet **MS Graph**, la fenêtre des variables - en ayant pris soin d'instancier l'objet - et éventuellement l'enregistreur de macros d'Excel à la recherche de nouvelles propriétés et méthodes. Rendez-vous dans un prochain chapitre pour aller toujours plus loin dans les profondeurs de **MS Graph**.

Retrouvez l'article de Fabrice Constans en ligne : [Lien 146](#)

Les derniers tutoriels et articles

Modélisation de XML : choix de structures pour la représentation des données

Quelques exemples de l'utilisation de XML en tant que structure de données.

1. Règles de création de documents XML

XML est utilisé à son profit maximum pour des flux/documents ayant une certaine temporalité ou à usages multiples. Sur des mécanismes de simple passage d'objet, on lui préférera d'autres technologies, comme JSON dans de nombreuses implémentations AJAX. Bien plus qu'un simple flux, il permet à son utilisateur de construire une structure de données dont l'utilisation pourra évoluer. Il est en ceci très proche des bases de données relationnelles où le schéma de la base se fonde, non sur l'utilisation qui en est faite mais sur la structure même de la donnée. S'il a le même avantage, il en a aussi les contraintes : toute construction d'un XML se doit de comporter une analyse minimum sur sa forme et les relations entre les données contenues. Les exemples qui suivent ne prétendent pas être exhaustifs mais donnent quelques pistes en la matière.

1.1. Exemple 1 : sémantique

Le nom des balises d'un flux XML se doit d'avoir un sens, d'être compréhensible par quelqu'un n'ayant qu'un minimum de connaissances fonctionnelles du sujet ; de même que dans une base de données relationnelle, on choisira des noms descriptifs pour les tables et leurs champs.

Exemple :

```
xml
<?xml version="1.0"?>
<a>
  <p>
    <pr>Claude</pr>
    <no>Bernard</no>
    <num>4475</num>
  </p>
</a>
```

On supposera que pr correspond au prénom et no au nom mais qu'en est-il de num ?

Est-ce un numéro de téléphone interne ? Un matricule ? Un code de bureau ?

Ce type d'erreur peut coûter très cher en maintenance ou en développement, en particulier lors de passage de connaissance. Un format XML est fait pour être utilisé par de nombreuses personnes, son sens doit être le plus clair possible.

```
xml
<?xml version="1.0"?>
<annuaire>
  <personne>
    <prenom>Claude</prenom>
    <nom>Bernard</nom>
    <numTel>4475</numTel>
  </personne>
</annuaire>
```

1.2. Exemple 2 : de l'usage des attributs

Un autre exemple, malheureusement fréquent, qu'il ne faut pas suivre : miser l'ensemble des informations sur le nom d'une balise.

En règle générale on considérera qu'un nom de balise est là pour indiquer la nature de l'information contenue, ses attributs pour la différencier des autres. La donnée brute est contenue dans la balise, les éléments permettant de la traiter dans les attributs.

Quelques exemples classiques d'utilisation d'attributs :

- identifiant ;
- langue du texte ;
- options d'affichage.

Par exemple à ceci :

```
xml
<personnes>
  <homme>..</homme>
  <femme>..</femme>
</personnes>
```

on préférera :

```
xml
<personnes>
  <personne sexe="M">..</personne>
  <personne sexe="F">..</personne>
</personnes>
```

Pour prendre un exemple moins caricatural et assez fréquent :

```
xml
<?xml version="1.0"?>
<annuaire>
  <personne>
    <prenom>Claude</prenom>
    <nom>Bernard</nom>
    <TelFixe>0136784478</TelFixe>
    <TelPort>0675789078</TelPort>
  </personne>
</annuaire>
```

Cette version risque en plus de devoir être modifiée, si la personne a un fixe personnel et professionnel. Verra-t-on alors apparaître TelFixePerso, TelFixePro ? Ceci posera un problème par la suite. En effet, dans tous les traitements liés au XML, de la validation à la transformation tout est centré sur l'élément et donc principalement sur son nom.

Rajouter un nouvel élément n'est donc jamais neutre.

Il est toujours préférable d'ajouter une nouvelle valeur d'attribut, celle-ci pouvant se permettre d'être plus exhaustive

```
xml
<?xml version="1.0"?>
<annuaire>
  <personne>
    <prenom>Claude</prenom>
    <nom>Bernard</nom>
    <Tel type="fixe personnel">0136784478</Tel>
    <Tel type="fixe professionnel">0136778945</Tel>
    <Tel type="portable personnel">0675789078</Tel>
  </personne>
</annuaire>
```

1.3. Exemple 3 : une structure ordonnée

Dans un XML l'ordre des éléments est une information en soi.

Contrairement au SGBDR et au SQL, où si aucun ordre n'est spécifié dans la requête, on ne peut prévoir l'ordonnement du résultat. Celui par défaut sur un XML est celui de sa déclaration dans le fichier texte (hormis le cas des attributs). Si on gère donc de façon pertinente les différents types de mise à jour d'un fichier XML, l'ordre du fichier devient une donnée et on n'est plus obligé de le coder via un élément ou un attribut.

```
xml
<?xml version="1.0"?>
<donnees>
  <ligne num="1">...</ligne>
  <ligne num="2">...</ligne>
  <ligne num="3">...</ligne>
  <ligne num="4">...</ligne>
</donnees>
```

Il faut savoir que les langages et outils de sélection sur XML permettent tous :

- de récupérer le *énième* élément ;
- de connaître le nombre d'éléments et/ou de sélectionner automatiquement le dernier.

De fait l'attribut **num** ici n'est indispensable que si trois conditions sont présentes :

- si l'ordre est une information pertinente ;
- si le fichier peut être modifié ;
- si la création/modification ne garantit pas une séquence valide des éléments.

1.4. Exemple 4 : hiérarchie et arborescence

Un des pires exemples d'utilisation XML est sans aucun doute celui où sa nature arborescente est inutilisée. Cette dernière permet en effet de stocker naturellement des informations de ce type (document, hiérarchie,

organisation d'entreprises) sans passer par les systèmes d'association des structures.

Continuer d'appliquer une solution de forme tabulaire à un document XML n'est malheureusement pas un cas rare mais aussi un de ceux qui posent les plus graves problèmes de traitements par la suite.

Un exemple trop fréquemment appliqué :

```
xml
<?xml version="1.0" encoding="UTF-8"?>
<livre>
  <chapitre num="1">
    <titre>A</titre>
    <corps>...</corps>
  </chapitre>
  <chapitre num="1.1">
    <titre>AA</titre>
    <corps>...</corps>
  </chapitre>
  <chapitre num="1.1.1">
    <titre>AAA</titre>
    <corps>...</corps>
  </chapitre>
  <chapitre num="1.1.2">
    <titre>AAB</titre>
    <corps>...</corps>
  </chapitre>
  <chapitre num="1.2">
    <titre>AB</titre>
    <corps>...</corps>
  </chapitre>
  <chapitre num="2">
    <titre>B</titre>
    <corps>...</corps>
  </chapitre>
  <chapitre num="2.1">
    <titre>BB</titre>
    <corps>...</corps>
  </chapitre>
</livre>
```

Cette donnée est naturellement arborescente. Cette forme séquentielle impose de lourds traitements pour pouvoir repasser dans un format arborescent. De plus, des instructions spécifiques, en particulier en XSLT (xsl:number) permettent l'équivalent de numérotation de l'attribut **num**. Dans le domaine de la donnée, toute donnée superflue ou répétée est équivalente à du bruit et introduit un risque d'erreur supplémentaire en cas de modification.

Le format d'un tel document devrait donc être au minimum :

```
xml
<?xml version="1.0" encoding="UTF-8"?>
<livre>
  <chapitre num="1">
    <titre>A</titre>
    <corps>...</corps>
  <chapitre num="1.1">
    <titre>AA</titre>
    <corps>...</corps>
  <chapitre num="1.1.1">
    <titre>AAA</titre>
    <corps>...</corps>
  </chapitre>
```

```

        <chapitre num="1.1.2">
            <titre>AAB</titre>
            <corps>....</corps>
        </chapitre>
    </chapitre>
    <chapitre num="1.2">
        <titre>AB</titre>
        <corps>....</corps>
    </chapitre>
</chapitre>
<chapitre num="2">
    <titre>B</titre>
    <corps>....</corps>
    <chapitre num="2.1">
        <titre>BB</titre>
        <corps>....</corps>
    </chapitre>
</chapitre>
</livre>

```

Mais comme nous l'avons vu dans la partie précédente, la numérotation des chapitres est redondante par rapport à la structure du fichier ci-dessous.

```

xml
<?xml version="1.0" encoding="UTF-8"?>
<livre>
    <chapitre>
        <titre>A</titre>
        <corps>....</corps>
    </chapitre>
    <chapitre>
        <titre>AA</titre>
        <corps>....</corps>
    </chapitre>
    <chapitre>
        <titre>AAA</titre>
        <corps>....</corps>
    </chapitre>
    <chapitre>
        <titre>AAB</titre>
        <corps>....</corps>
    </chapitre>
    <chapitre>
        <titre>AB</titre>
        <corps>....</corps>
    </chapitre>
</livre>

```

2. Utilisations de multiples XML

Au niveau des traitements, que ce soit DOM, XQuery (langage de requêtes) ou XSLT (langage de transformation), tous permettent d'atteindre différents fichiers ainsi que des mécanismes de sélection. Il peut donc être plus pertinent d'opérer une partition des données en fonction des problématiques.

2.1. Problématique du XML unique

Dans une application vous n'êtes pas obligé de stocker toutes vos données dans un seul fichier XML.

Un fichier unique posera différents problèmes, entre autres :

- problème de volumétrie si le travail nécessite un chargement en mémoire ;
- mise à jour, lecture et autres traitements plus longs (toujours fonction de la taille mémoire occupée) ;
- sécurité des données, tous vos utilisateurs n'ont pas forcément les mêmes niveaux de visibilité ;
- manque de lisibilité et complexité accrue, en particulier si différents ensembles fonctionnels y sont regroupés.

2.2. Organisation des XML

Par rapport aux problèmes évoqués précédemment certains doivent être traités prioritairement.

En priorité, gérer les problèmes d'accès sécurité s'ils existent. Il est difficile et pas forcément fiable d'essayer de limiter l'accès d'une application à une partie d'un fichier.

Ensuite il vous faudra gérer l'aspect fonctionnel de vos données. Comme pour un SGBDR, cela revient à réorganiser en priorité vos données, non selon l'utilisation que vous en faites à un instant t mais selon leur nature et leurs relations.

Enfin, s'il existe encore, gérer le problème volumétrique.

Si on traite ce problème en dernier c'est qu'il peut, en plus d'être réglé par les deux précédentes réorganisations, être soulagé par une solution technologique. En effet, de nombreuses solutions s'orientent vers des solutions partiellement SAX ou orientées streaming, ce qui permet de moins se préoccuper de la taille. Si cela ne convient pas, il faut s'inspirer des solutions de partitionnement sur les SGBD. Trouver une distinction qui soit aussi physique que logique : année, mois, entreprise, produit... Il existe des types d'API comme Cocoon qui utilisent des technologies comme Xlink, permettant de gérer ces fichiers comme des ensembles.

Si vous appliquez ces traitements et que vous voyez exploser le nombre de vos fichiers XML il est peut-être temps de regarder du côté d'une solution base de données XML native ou au moins de les intégrer à un SGBDR comme le permettent Oracle, DB2, SQL Server...

Retrouvez l'article d'Erwan Amoureux en ligne : [Lien 147](#)

Présentation du langage XML - La galaxie XML : le langage et ses dérivés

Présentation du langage XML et des technologies liées. Un premier point sera fait sur le format, quelques règles d'utilisation puis un aperçu des technologies afférentes.

1. Format XML

1.1. Notions générales

1.1.1. Historique

XML est une recommandation du W3C, 10 février 1998. Il est né du succès du HTML mais aussi de la constatation de ses insuffisances. Comme ce dernier, XML est issu du langage de balise SGML et en est une simplification. Là où HTML est spécialisé dans l'affichage, XML lui se spécialise dans la donnée.

1.1.2. Caractéristiques

Si nous reprenons sa dénomination, eXtensible Markup Language :

- Language : interprété, notion de sémantique ;
- Markup : balise, les symboles utilisés pour délimiter les mots de ce langage ;
- extensible : métalangage, équivalent à une liste de phonèmes, plus des règles de ponctuation.

Les objectifs de conception de XML, listés dans sa spécification, sont les suivants :

1. XML devrait pouvoir être utilisé sans difficulté sur Internet ;
2. XML devrait soutenir une grande variété d'applications ;
3. XML devra être compatible avec SGML ;
4. il devrait être facile d'écrire des programmes traitant les documents XML ;
5. le nombre d'options dans XML doit être réduit au minimum, idéalement à aucune ;
6. les documents XML devraient être lisibles par l'homme et raisonnablement clairs ;
7. la conception de XML devrait être préparée rapidement ;
8. la conception de XML sera formelle et concise ;
9. il devrait être facile de créer des documents XML ;
10. la concision dans le balisage de XML est de peu d'importance.

1.1.3. Usages

Une liste non exhaustive de ses différentes utilisations :

- formats d'échange/extraction de données (une des plus anciennes) ;
- formats de document (open XML, openDocument, RSS, XBRL...);
- formats de langage descriptif : XHTML, math ml, svg... ;
- sauvegarde de données (du fichier de configuration à la base de données XML native...).

1.2. Éléments constitutifs d'un XML

1.2.1. L'en-tête

Un document XML devrait toujours commencer par un en-tête.

L'en-tête minimum :

```
<?xml version="1.0"?>
```

version représente la version de XML utilisée

NB: il existe d'autres versions que la 1.0 mais quasi inutilisées.

Cet en-tête peut être complété par l'encodage qui est utilisé pour la lecture

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

Si cet encodage n'est pas précisé UTF-8 sera utilisé par défaut.

Un autre élément facultatif **standalone**

Cet élément signale si le XML en lui seul est suffisant pour être lu ou s'il nécessite l'adjonction d'un autre document externe (DTD). Ce point sera abordé plus en détail dans la partie des entités.

La valeur par défaut de **standalone** est yes

```
<?xml version="1.0" standalone='yes'?>
```

1.2.2. Les balises, appelées éléments

Les balises, dites éléments, sont l'autre partie obligatoire d'un XML, c'est le type de base du XML, celui qui va permettre de créer l'arborescence.

Pour tout élément avec contenu, une balise de début et de fin sont obligatoires.

```
<nom_balise>contenu_balise</nom_balise>
```

Pour un élément sans contenu on peut utiliser une écriture raccourcie

Élément vide

```
<nom_balise/>
```

le nom de la balise suit aussi des règles :

pour le premier caractère on autorise uniquement : les lettres (même accentuées), '.' et '_' ;

pour les caractères suivants on ajoute à cette liste : les chiffres, '-' et '+' ;

un élément peut soit contenir du texte, soit d'autres éléments soit un mélange des deux (le XHTML est un bon exemple de ce dernier cas) ;

pour ce qui est du texte certains caractères sont interdits. Ainsi '<' et '>' seront remplacés par leur référence < et > ;

à la suite de l'en-tête, on ne doit trouver qu'**une unique balise** qu'on appellera **élément racine**.

Ainsi ce code est **faux** :

```
xml
<?xml version="1.0"?>
<racine>
  ...
</racine>
<racine2>
  ...
</racine2>
```

le XML minimum serait donc de la forme :

```
xml
<?xml version="1.0"?>
<ElementRacine/>
```

Les balises ne peuvent se superposer.

Ce code est exact :

```
xml
<?xml version="1.0"?>
<racine>
  <a>
    <b/>
    <c/>
  </a>
</racine>
```

Ce code est faux :

```
xml
<?xml version="1.0"?>
<racine>
  <a>
    <b>
    <c>
    </b>
    </c>
  </a>
</racine>
```

1.2.3. Attribut

Un élément peut être porteur d'un ou plusieurs attributs.

```
xml
<Element att1="test1" att2="test2">...</Element>
```

Ici Element est porteur de deux attributs att1 et att2.

Les noms des attributs suivent les mêmes règles d'écriture que ceux des éléments.

Les attributs ne peuvent contenir que du texte.

Un élément ne peut posséder deux attributs de même nom, ainsi le code suivant est faux

```
xml
<Element att1="test1" att1="test2">...</Element>
```

1.2.4. namespace

Les namespaces (espaces de nom) sont la solution trouvée au problème des balises homonymes.

En effet, XML étant un métalangage, deux concepteurs peuvent avoir fait les mêmes choix de noms de balises sans que leurs interprétations soient identiques. Il est en effet fréquent de trouver des XML composés de plusieurs langages ou formats.

Un namespace se présente sous la forme d'un « attribut » xmlns (XML namespace) suivi d'une URI.

Un namespace peut se déclarer tel quel, dans ce cas, toutes les balises contenues dans la balise porteuse, et qui ne sont pas porteuses d'un autre namespace, seront considérées comme incluses dans l'espace de nom.

Soit :

```
xml
<?xml version="1.0"?>
<racine>
  <a xmlns="Nstest1">
    <b>
      <c/>
    </b>
    <d xmlns="Nstest2"/>
  </a>
  <e/>
</racine>
```

Les balises a, b, c appartiennent à l'espace de nom Nstest1.

La balise d appartient à l'espace de nom Nstest2.

Les balises **racine** et **e** n'appartiennent à aucun de ces deux namespaces. Deuxième façon de procéder, l'utilisation d'un préfixe. Dans la déclaration du namespace on fait alors suivre xmlns du préfixe choisi. Toutes les balises contenues dans la balise où il est déclaré (y compris cette dernière), porteuses du préfixe, appartiendront alors au namespace. L'usage et la commodité veulent que ce soit dans le premier élément du XML qu'apparaissent les namespaces.

```
xml
<?xml version="1.0"?>
<racine xmlns:t1="Nstest1" xmlns:t2="Nstest2">
  <t1:a>
    <b>
      <t1:c/>
    </b>
    <d/>
  </t1:a>
  <t2:e/>
</racine>
```

les balises **a** et **c** appartiennent à l'espace de nom **Nstest1**.

la balise **e** appartient à l'espace de nom **Nstest2**.

les balises **racine**, **b** et **d** n'appartiennent à aucun de ces deux namespaces.

1.2.5. CDATA

Plutôt que de modifier un par un les '&', '<' et '>' du texte d'une balise, on peut utiliser les CDATA. Tout '&', '<' ou '>' contenu dans un CDATA est alors interprété comme un `&`, `<` ou `>`;

```
xml
<?xml version="1.0"?>
<racine><![CDATA[exemple du contenu d'un CDATA
avec des < et >]]></racine>
```

Attention, toute balise placée dans un CDATA sera considérée comme du texte.

Si on est face à une balise à contenu mixte il faudra donc opter pour une solution de ce type :

```
xml
<?xml version="1.0"?>
<racine>
<![CDATA[exemple du contenu d'un CDATA avec des <
et >]]>
<a>...</a>
<![CDATA[suite avec < et >]]>
</racine>
```

1.2.6. Commentaires

Comme en HTML, il est possible de faire des commentaires dans du XML.

Ceux-ci doivent se situer après l'en-tête. Ils peuvent être imbriqués entre des éléments exactement comme des balises.

Un commentaire commence par `<!--` et se termine par `-->`.

```
xml
<?xml version="1.0"?>
<!--exemple de commentaires après l'en-tête-->
<racine>
  <!--exemple de commentaires imbriqués dans
des éléments-->
  <a>...</a>
  <b>...</b>
</racine>
<!--exemple de commentaires après l'élément
racine-->
```

1.2.7. processing-instruction

Les processing-instructions sont le moyen d'envoyer des ordres à l'outil qui lit le XML, généralement le parseur.

Une processing-instruction commence par `<?` et se termine par `>`. L'en-tête est un exemple de processing-instruction.

Plus loin dans le cours, on les utilisera aussi pour lier une feuille XSLT à un XML.

1.2.8. Les entités

La norme XML a entre autres comme but l'échange de données, elle nécessite un outil de validation. Elle a hérité en ce sens du SGML des DTD (document type declaration) et comme SGML, une DTD peut être incluse à ce titre dans le XML.

Pour qu'un XML soit bien formé, la DTD doit impérativement être déclarée avant le premier élément, comme suit.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE greeting [
  <!ELEMENT greeting (#PCDATA)>
]>
<greeting>Hello, world!</greeting>
```

Le mécanisme des DTD permet l'utilisation des entités références. Une entité références fonctionne comme une mutalisation de chaîne de caractères. Exemple :

```
<?xml version="1.0"?>
<!DOCTYPE racine [
  <!ENTITY hel "hello world!!!">
]>
<racine>&hel;</racine>
```

est équivalente à :

```
<?xml version="1.0"?>
<racine>hello world!!!</racine>
```

L'attribut **standalone** de l'en-tête vu précédemment doit être à **no** si le XML utilise ce système d'entité avec une DTD externe.

Je ne saurais que vivement déconseiller ceci et en particulier l'utilisation des entités externes. Conçu bien avant l'explosion du Net, il comporte d'énormes failles de sécurité qui rendent aisées des attaques par déni de service. Voir Sécurité applicative : Attaque de Déni de Service via DTD/XML pour plus de détails : [Lien 147](#).

1.3. Récapitulatif sur la construction d'un XML

Les principales règles sur un XML sont :

1. un en-tête indiquant la version (certains outils la considérant obligatoire mieux vaut toujours la servir) ;
2. un seul élément racine ;
3. toute balise ouverte doit être refermée ;
4. pas de recouvrement de balises ;
5. une balise ne peut pas posséder deux attributs de même nom ;
6. les '&', '<' et '>' non compris dans un CDATA doivent être remplacés par `&`, `<` et `>`.

Un document respectant toutes les règles de construction est dit bien formé, il pourra alors utiliser tous les outils à la disposition de XML.

2. La galaxie XML

Un des termes descriptifs utilisés pour les technologies XML est celui de galaxie avec XML en son centre et différentes étoiles voire amas qui s'éloignent suivant une logique de cercles.

Les détracteurs de XML pourront sans aucun doute faire remarquer que c'est une description qui se rapprocherait plus de celle de l'Enfer de Dante.

Au centre
XML

Premier Cercle
parcours/lecture : parseur(DOM, SAX...)

deuxième Cercle
La grammaire/validation : DTD, XML Schema, Relax NG...

Troisième Cercle
La sélection : Xpath

La transformation et requête : XSLT, XQUERY, DOM

Quatrième Cercle
les liens : Xlink, Xinclude, XPointer...

Cinquième Cercle
Producteur/porteur de XML : Xforms, XSQL, SOAP

Sixième Cercle
Les langages descriptifs de format XML : Open document, SVG, RSS, MathML, XBRL...

Septième Cercle
Métalangage au service des métadonnées : RDF

2.1. Les parseurs

Un XML, c'est avant tout un fichier ou un flux texte. S'il doit être utilisé par un langage, celui-ci doit choisir la façon de le stocker en mémoire et de le parcourir, c'est le rôle d'un parseur.

Cet outil logiciel, parfois un analyseur syntaxique, permet à une application cliente de valider (s'il est validant) un document XML et de le lire, voire de le modifier. Un parseur met à disposition de l'application cliente les données XML lues au travers d'API, les plus répandues étant SAX et DOM. Une autre catégorie, plus récente, intègre une technologie PULL, associant les avantages de SAX - définie comme une technologie PUSH - et DOM sans leurs inconvénients.

2.1.1. DOM

DOM (*Document Object Model*) est une API permettant d'accéder au contenu d'un document XML sous la forme d'une structure arborescente. Le document XML, après avoir été totalement chargé en mémoire, est accessible au travers d'un ensemble d'objets correspondant aux différents types de nœuds qui s'y trouvent, et exposant les méthodes permettant de parcourir l'arbre, de façon hiérarchique ou transversale. L'approche diffère donc

totalement de celle de SAX, bien qu'un arbre DOM puisse être construit en utilisant ce dernier. DOM est une recommandation du W3C, proposée en plusieurs versions (level) proposant des fonctionnalités croissantes et dont la compatibilité est ascendante ; Le level 1 est devenu une recommandation le 1er octobre 98, le level 2 (XML) le 13 novembre 2000 et tous les derniers éléments du level 3 sont des recommandations depuis avril 2004. Il existe des implémentations de DOM dans pratiquement tous les langages interprétés ou compilés existants pouvant lire des documents XML.

2.1.2. SAX

SAX (*Simple API for XML*) est une API basée sur un modèle événementiel, qui transforme un document XML en un flux d'événements déclenchés par la lecture d'éléments syntaxiques XML (balise ouvrante, balise fermante, etc.). Le modèle est quelque peu calqué sur celui des interfaces graphiques, l'application cliente devenant un "écouteur d'événement". SAX a comme

avantage, grâce à son fonctionnement, de ne lire le code que par petites portions, ce qui lui évite de le charger en mémoire intégralement, contrairement à DOM. Il est donc très intéressant pour la lecture des gros documents. Il ne sera par contre pas adapté aux cas où l'application cliente ne peut pas se contenter d'un parcours linéaire et par "petits bouts" du document XML.

2.1.3. Autres

STAX (*Streaming API for XML*) est un exemple d'API Java Pull implémentée par des parseurs de BEA et Sun.

<http://www.jcp.org/en/jsr/detail?id=173>

De plus le modèle objet de DOM a subi de nombreuses critiques : trop lourd, trop complexe et ne satisfaisant pas une optique objet.

Il existe donc de nombreuses bibliothèques, particulièrement en JAVA (JDOM) et en PHP

(SimpleXML...) qui tout en s'appuyant sur ces modèles offrent des interfaces très différentes.

2.2. La validation

En décrivant au début de ce cours les règles du format XML, nous avons décrit les conditions nécessaires pour qu'un XML soit bien formé. C'est la condition indispensable pour qu'un document puisse utiliser les différents outils du monde XML, il n'est autrement qu'un fichier texte rempli de balises.

Néanmoins ceci reste insuffisant, en effet qu'un document XML soit utilisé comme un langage ou comme un format d'échange, il est soumis dans chacun des cas à des contraintes syntaxiques.

En règle générale les principales contraintes sont :

- nom des éléments et des attributs autorisés ;
- hiérarchie(père/fils) et appartenance des éléments et attributs ;
- séquençement et itérations des éléments ;
- contraintes d'unicité et de référence.

C'est le parseur, à l'aide d'un document décrivant la grammaire exigée, qui se chargera de ce travail ; on parle alors de parseur validant. La validation consiste donc à comparer, lors de la lecture d'un document XML par le parseur validant, la structure du fichier XML avec la grammaire fournie. On considérera le XML valide si les deux concordent.

Les principaux langages de validation sont :

- les DTD : c'est le plus ancien, il est antérieur au XML, c'est aussi un des plus restreints en termes de contraintes pouvant être exprimées ;
- XML Schema : la norme syntaxique fournie par le W3C, permet un très grand choix de contraintes et de types mais souvent considéré comme trop complexe. Sans doute le langage le plus utilisé avec les DTD ;
- Relax NG : l'équivalent de XML Schema mais proposé par OASIS. Il sert de base au document de Open Office. Plus souple que XML Schema mais néanmoins moins implémenté ;
- Schematron : schematron n'est pas un langage validant «complet», il est généralement utilisé en complément d'un des précédents cités. Là où les autres sont principalement orientés vers des contraintes structurelles, lui est plus orienté « sémantique » en permettant d'en ajouter sur les valeurs des éléments et des attributs.

2.3. Sélection, transformation, requête

La sélection :

XPath est un langage avec une syntaxe non XML, permettant d'adresser les différents nœuds ou groupes de nœuds particuliers d'un document XML.

Contrairement à un langage de sélection comme SQL, il ne permet pas, en dehors de certaines opérations arithmétiques ou concernant les chaînes, de transformer la structure des données.

La transformation :

XSLT : langage de forme XML. Basé sur un système de templates, il permet de transformer un flux XML soit en un autre flux XML, soit HTML ou simplement texte

La requête :

Xquery : langage qui n'utilise pas la forme XML. Il permet à partir d'un ou plusieurs documents XML de reconstruire un nouveau XML.

2.4. Les liens

Xlink, Xpointer : essentiellement basés sur Xpath et un ensemble d'attributs, ils permettent de créer des liens entre différents éléments d'un même document ou au travers

d'autres. Ces technologies sont peu utilisées et implémentées isolément. On les retrouve surtout au sein d'API comme COCOON traitant du XML ou de langages comme XBRL.

2.5. Echanges XML

SOAP : protocole RPC, particulièrement utilisé dans les Web Services. Il consiste en un fichier XML en deux parties :

- l'enveloppe : qui contient les informations sur le message lui-même ;
- un modèle de données : les informations transmises.

XSQL : langage développé par Oracle. Il consiste en un fichier de format XML contenant une requête SQL dans une première partie et la présentation XML désirée des données récupérées dans la seconde. Il est atteignable via des requêtes HTTP.

XFORMS : un nouveau type de formulaires pour le Web. En découpant les formulaires traditionnels XHTML en trois parties — le modèle XForms, les données d'instance et l'interface d'utilisateur*— XForms sépare la présentation du contenu, permet la réutilisation, apporte un typage fort, tout en réduisant les allers-retours au serveur comme en offrant une indépendance par rapport au matériel et en limitant les recours à des scripts. Il permet l'envoi préformatés de données soit sous forme d'une string, soit sous la forme d'un XML.

2.6. Langages descriptifs

XML est principalement utilisé dans des langages descriptifs, quelques exemples parmi la multitude :

XHTML : version XML du format HTML ;

OpenDocument : format XML utilisé par OOo ;

OpenXml : format utilisé par la suite Microsoft Office ;

RSS : syndication de contenus web ;

SVG : format de description d'image vectorielle ;

MathML : format de description de formules mathématiques ;

XBRL : format de description de données financières ;

2.7. Métadonnées

RDF,OWL: format de liens de métadonnées. Ils permettent d'indexer et de renseigner des structures d'informations de natures et formats divers, et de permettre par là un traitement applicatif

Retrouvez l'article d'Erwan Amoureux en ligne : [Lien 149](#)

Liens

- Lien 01 : <http://www.jboss.org/jbossas/downloads.html>
- Lien 02 : <http://www.developpez.net/forums/d1020147/java/serveurs-conteneurs-java-ee/jboss/jboss-application-server-6-disponible/>
- Lien 03 : <http://www.developpez.com/actu/27476/Index-TIOBE-Python-sacre-langage-de-l-annee-Java-toujours-numero-1/>
- Lien 04 : <http://apache.developpez.com/actu/24953/La-Fondation-Apache-quitte-le-Java-Community-Process-le-projet-Harmony-pourrait-etre-abandonne/>
- Lien 05 : <http://www.developpez.net/forums/d1029185/java/general-java/oracle-va-t-cantonner-futur-java-aux-besoins-grandes-entreprises/>
- Lien 06 : <http://www.developpez.net/forums/d1027966/club-professionnels-informatique/actualites/exploits-java-ont-depasse-ciblant-produits-adobe-2010-dapres-cisco/>
- Lien 07 : <http://www.jpackages.com/>
- Lien 08 : <http://www.jpackages.com/jflashplayer/>
- Lien 09 : <http://www.jpackages.com/jflashplayer/download/>
- Lien 10 : <http://www.jpackages.com/docs/jflashplayer/apispec/>
- Lien 11 : <http://www.adobe.com/fr/downloads/>
- Lien 12 : http://fr.wikipedia.org/wiki/Adobe_Flash
- Lien 13 : <ftp://ftp-developpez.com/slim-boukettaya/tutoriels/afficher-animations-flash-dans-composant-swing/animation%20Flash.rar>
- Lien 14 : <http://slim-boukettaya.developpez.com/tutoriels/java-et-swf/>
- Lien 15 : <http://slim-boukettaya.developpez.com/tutoriels/afficher-animations-flash-dans-composant-swing/>
- Lien 16 : <http://github.com/cyrilmottier/GreenDroid>
- Lien 17 : <http://cyril-mottier.developpez.com/tutoriels/android/introduction-styles-themes/fichiers/StylesAndThemes.zip>
- Lien 18 : <http://developer.android.com/intl/fr/reference/android/R.attr.html>
- Lien 19 : <http://cyril-mottier.developpez.com/tutoriels/android/introduction-styles-themes/>
- Lien 20 : http://www.zenika.com/conference/java/concurrent_programming_with_spring_and_david_syer
- Lien 21 : <http://www.springsource.org/>
- Lien 22 : <http://blog.springsource.com/2007/11/18/interface21-becomes-springsource/>
- Lien 23 : <http://www.vmware.com/>
- Lien 24 : <http://static.springsource.org/spring-batch/>
- Lien 25 : <http://www.springsource.org/spring-integration>
- Lien 26 : <http://www.springsource.org/spring-amqp>
- Lien 27 : <http://static.springsource.org/spring-batch-admin/getting-started.html>
- Lien 28 : <http://manning.com/templier/>
- Lien 29 : http://www.zenika.com/experts/arnaud_cogoluegnes?fg=50011
- Lien 30 : http://www.zenika.com/formation_enterprise_integration_avec_spring.php?fg=50011
- Lien 31 : <http://www.zenika.com/?fg=50011>
- Lien 32 : <http://www.activiti.org/team.html>
- Lien 33 : <http://www.slideshare.net/DaveSyer/concurrency-fisher-syer-s2gx-2010>
- Lien 34 : <http://www.developpez.net/forums/redirect-to/?redirect=http%3A%2F%2Fgit.springsource.org%2Fs2gx-2010%2Fconcurrent-programming-distributed-applications>
- Lien 35 : <http://longbeach.developpez.com/interviews/Spring/DavidSyer/>
- Lien 36 : <http://www.zend.com/community/downloads>
- Lien 37 : <http://www.wampserver.com/download.php>
- Lien 38 : <http://sourceforge.net/projects/wampserver/files/WampServer%202/WampServer%202.0/WampServer2.0i.exe/download>
- Lien 39 : <http://www.wampserver.com/phorum/list.php?1>
- Lien 40 : <http://m-vaudin.developpez.com/tutoriels/zend-framework/installation-creation-projet/>
- Lien 41 : <http://jasperforge.org/projects/jasperserver>
- Lien 42 : <http://business-intelligence.developpez.com/livre-blanc-bi-open-source/>
- Lien 43 : <http://ssrsphp.codeplex.com/>
- Lien 44 : <http://framework.zend.com/manual/fr/learning.quickstart.create-project.html>
- Lien 45 : <http://framework.zend.com/manual/fr/zend.controller.actionhelpers.html>
- Lien 46 : <http://ovanhoof.developpez.com/phpssrs/>
- Lien 47 : <http://debray-jerome.developpez.com/demos/imageronde.html#demo1>
- Lien 48 : <http://debray-jerome.developpez.com/demos/imageronde.html#demo2>
- Lien 49 : <http://debray-jerome.developpez.com/demos/imageronde.html#demo3>
- Lien 50 : <http://debray-jerome.developpez.com/articles/creer-une-image-ronde-en-css3/>
- Lien 51 : <http://debray-jerome.developpez.com/demos/multiplebackground.html#demo1>
- Lien 52 : <http://debray-jerome.developpez.com/demos/multiplebackground.html#demo2>
- Lien 53 : <http://debray-jerome.developpez.com/articles/multiples-backgrounds-en-css3/>
- Lien 54 : <http://debray-jerome.developpez.com/demos/bordures.html#demo1>
- Lien 55 : <http://debray-jerome.developpez.com/demos/bordures.html#demo2a>
- Lien 56 : <http://debray-jerome.developpez.com/demos/bordures.html#demo2b>
- Lien 57 : <http://debray-jerome.developpez.com/demos/bordures.html#demo3>
- Lien 58 : <http://debray-jerome.developpez.com/demos/bordures.html#demo4>
- Lien 59 : <http://debray-jerome.developpez.com/articles/les-bordures-en-css3/>
- Lien 60 : <http://braincracking.org/>
- Lien 61 : <http://braincracking.org/2010/11/24/tester-fiablement-ses-navigateurs/>
- Lien 62 : <http://yui-blog.com/blog/2010/11/03/gbs-update-2010q4/>
- Lien 63 : http://en.wikipedia.org/wiki/United_States_v._Microsoft
- Lien 64 : <http://www.my-debugbar.com/wiki/IETester/HomePage>
- Lien 65 : <http://utilu.com/IECollection/>
- Lien 66 : http://tredosoft.com/Multiple_IE
- Lien 67 : <http://braincracking.org/2009/11/27/cest-comme-ca-quon-fixe-les-bugs-chez-ms/>
- Lien 68 : <http://www.redmondpie.com/how-to-install-os-x-snow-leopard-in-vmware-windows-7-9140301/>
- Lien 69 : <http://www.apple.com/fr/macbookair/features.html#flash-storage>
- Lien 70 : <http://www.microsoft.com/downloads/en/details.aspx?FamilyID=21eabb90-958f-4b64-b5f1-73d0a413c8ef&displaylang=en>
- Lien 71 : <http://www.parallels.com/fr/products/desktop/>
- Lien 72 : <http://www.vmware.com/products/fusion/>
- Lien 73 : <http://www.virtualbox.org/>
- Lien 74 : <http://www.microsoft.com/windows/virtual-pc/default.aspx>

Lien 75 : <http://developer.yahoo.com/yui/articles/gbs/>
 Lien 76 : <http://www.realvnc.com/vnc/index.html>
 Lien 77 : <http://jpvincent.developpez.com/tutoriels/web/tester-fiablement-ses-navigateurs/>
 Lien 78 : <http://fr.wikipedia.org/wiki/htaccess>
 Lien 79 : <http://marc-chappuis.developpez.com/tutoriels/ajax/ajax-et-referencement/fichiers/ajax-referencement.zip>
 Lien 80 : <http://marc-chappuis.developpez.com/tutoriels/ajax/ajax-et-referencement/>
 Lien 81 : <http://perl.developpez.com/faq/tk/?page=Canevas#GradientColor>
 Lien 82 : <http://perl.developpez.com/faq/tk/?page=Boutons#ColoredButton>
 Lien 83 : <http://perl.developpez.com/livres/#L9782841770816>
 Lien 84 : <http://perl.developpez.com/faq/tk/?page=HabillageComple#Palette>
 Lien 85 : <http://search.cpan.org/search?query=Tk%3A%3ACmdLine&mode=all>
 Lien 86 : <http://search.cpan.org/search?query=Tk%3A%3ADressing&mode=all>
 Lien 87 : <http://perl.developpez.com/faq/tk/>
 Lien 88 : <http://www.developpez.com/actu/22690>
 Lien 89 : <http://www.developpez.net/forums/d1018659/c-cpp/bibliotheques/qt/canonical-nokia-pourraient-travailler-collaboration-qt/>
 Lien 90 : <http://elixir.freebox.fr/>
 Lien 91 : <http://www.developpez.net/forums/d1012723/c-cpp/bibliotheques/qt/freebox-v6-developpee-qml/>
 Lien 92 : <http://www.developpez.net/forums/f172/autres-langages/python-zope/gui/pyside-pyqt/>
 Lien 93 : <http://pyqt.developpez.com/tutoriels/>
 Lien 94 : <http://pyqt.developpez.com/telecharger/>
 Lien 95 : <http://pyqt.developpez.com/livres/>
 Lien 96 : <http://pyqt.developpez.com/>
 Lien 97 : <http://www.developpez.net/forums/d1026882/autres-langages/python-zope/gui/pyside-pyqt/contribuez/ouverture-rubrique-pyqt-pyside/>
 Lien 98 : <http://qt.developpez.com/evenement/2010-devdays/#LIV-D-5>
 Lien 99 : <http://www.developpez.com/actu/21644/Qt-Mobility-1-1-est-sorti-en-beta-les-nouvelles-API-pour-mobiles-pour-le-framework-Qt-offriront-un-meilleur-support-pour-MeeGo>
 Lien 100 : <http://qt.developpez.com/evenement/2010-devdays/reportage/>
 Lien 101 : <http://labs.qt.nokia.com/2010/12/02/velvet-and-the-qml-scene-graph/>
 Lien 102 : http://labs.trolltech.com/page/Main_Page
 Lien 103 : <http://labs.qt.nokia.com/author/gunnar/>
 Lien 104 : <http://www.openscenegraph.org/projects/osg>
 Lien 105 : <http://qt.gitorious.org/qt-labs/scene-graph>
 Lien 106 : <http://qt-labs.developpez.com/graphics-performance/10-velours-qml-scene-graph/>
 Lien 107 : <http://developer.qt.nokia.com/>
 Lien 108 : http://developer.qt.nokia.com/wiki/PySide_Binding_Generation_Tutorial
 Lien 109 : <http://pyside.org/files/binding-tutorial.tar.gz>
 Lien 110 : <http://doc.trolltech.com/qtjambi-4.4/html/com/trolltech/qt/qtjambi-typesystem.html>
 Lien 111 : <http://www.pyside.org/docs/apiextractor/typesystem.html>
 Lien 112 : <http://doc.trolltech.com/qtjambi-4.4/html/com/trolltech/qt/qtjambi-typesystem.html#using-code-templates>
 Lien 113 : <http://qt.gitorious.org/pyside/pyside>
 Lien 114 : <http://qt-devnet.developpez.com/tutoriels/python/pyside/binding-shiboken/>
 Lien 115 : <http://www.microsoft.com/fr-fr/library/ms668604.aspx>
 Lien 116 : <http://create.msdn.com/en-us/resources/downloads>
 Lien 117 : <http://msdn.microsoft.com/fr-fr/library/ms668604.aspx>
 Lien 118 : <http://www.mathieu-perrein.net/>
 Lien 119 : <http://blog.mathieu-perrein.net/post/WP7-Creez-votre-compte-Marketplace-%21.aspx>
 Lien 120 : <http://blogs.msdn.com/b/pierrec/archive/2010/12/04/windows-phone-7-inscription-sur-marketplace-les-ressources-pour-s-en-sortir-dans-toutes-les-situations.aspx>
 Lien 121 : <http://www.e-naxos.com/DnlManager.aspx?GROUP=18&FILEID=101&Lang=FR>
 Lien 122 : <http://windowsphone.frogz.fr/>
 Lien 123 : <http://blogs.msdn.com/b/pierrec/>
 Lien 124 : <http://www.galasoft.ch/mvvm/getstarted/>
 Lien 125 : <http://labigne.over-blog.com/ext/http://gruaz.net/?p=293>
 Lien 126 : <http://www.aymericlagier.com/wp-content/uploads/2010/12/Bookmarks.zip>
 Lien 127 : <http://a-lagier.developpez.com/tutoriels/windowsphone/bien-commencer-developper-pour-windows-phone-7/>
 Lien 128 : <http://www.apple.com/fr/mac/app-store/>
 Lien 129 : <http://www.apple.com/fr/macosx/lion/>
 Lien 130 : <http://kaelisoft.developpez.com/tutoriels/mac/app-store/>
 Lien 131 : <http://dico.developpez.com/html/1713-Generalites-POO-Programmation-Orientee-Objet.php>
 Lien 132 : <http://dico.developpez.com/html/961-Langages-Garbage-Collector.php>
 Lien 133 : <http://a-renouard.developpez.com/tutoriels/ios/base-objective-c/>
 Lien 134 : http://www.youtube.com/watch?v=l_GJMXdl55A&feature=player_embedded
 Lien 135 : <http://www.winktoolkit.org/?section=previews&previews=63>
 Lien 136 : <http://www.winktoolkit.org/?section=previews&previews=47>
 Lien 137 : <http://www.winktoolkit.org/?section=previews&previews=112>
 Lien 138 : <http://www.winktoolkit.org/?section=previews&previews=51>
 Lien 139 : <http://www.winktoolkit.org/?section=previews&previews=55>
 Lien 140 : <http://www.winktoolkit.org/>
 Lien 141 : http://www.winktoolkit.org/download/wink_1.2_optimized.zip
 Lien 142 : <http://www.winktoolkit.org/?section=tutorials&tutorials=32>
 Lien 143 : <http://www.winktoolkit.org/tutorials/>
 Lien 144 : <http://www.winktoolkit.org/discussion/topic/get-involved-with-wink>
 Lien 145 : <http://wink.developpez.com/tutoriels/javascript/developper-votre-webapp-mobile-avec-wink/>
 Lien 146 : <http://loufab.developpez.com/tutoriels/access/msgraph2/>
 Lien 147 : <http://erwy.developpez.com/tutoriels/xml/modelisation-xml-choix-structures-pour-representation-donnees/>
 Lien 148 : <http://www.developpez.net/forums/d835909/autres-langages/xml-xsl-soap/securite-applicative-attaque-deni-service-via-dtd-xml-toutes-plateformes/>
 Lien 149 : <http://erwy.developpez.com/tutoriels/xml/galaxie-xml-langage-et-ses-derives/>