



# Developpez

*Le Mag*

Édition de décembre – janvier 2014/2015

Numéro 55

Magazine en ligne gratuit

Diffusion de copies conformes à l'original autorisée

Réalisation : Alexandre Pottiez & Sébastien Lataix

Rédaction : la rédaction de Developpez

Contact : magazine@redaction-developpez.com

## Sommaire

TypeScript	Page	2
2D/3D/Jeux	Page	18
Libres & Open Source	Page	25
Perl	Page	55
OpenOffice-LibreOffice	Page	62
Java	Page	71
Qt	Page	79
CRM	Page	83

## Éditorial

Afin d'assouvir votre soif, toujours plus grande de savoir, voici un nouveau numéro du magazine qui vous fera découvrir le meilleur de Developpez.com. Profitez-en bien !

La rédaction

## Article Perl



### Encodage de caractères avec Perl

Cet article décrit les différents encodages de caractères, comment ils peuvent conduire à des problèmes et comment les gérer dans un programme Perl.

par **Christophe Louvet**

Page 55



## Article OpenOffice-LibreOffice

### Créer un dépliant avec Texte (Writer)

Il existe des modèles pour réaliser des dépliants, mais découvrez comment en créer un par vous-même.

par **Vincent Viale**

Page 62



# TypeScript

## Les derniers tutoriels et articles

# Les unions de types en TypeScript : Spécifications provisoires

Cet article présente le concept puissant des unions de types. Ce sont les spécifications en cours d'implémentation (au moment de la rédaction de l'article) et qui pourraient rendre TypeScript assez unique dans le paysage des langages avec typage.

## 1 Avant-propos

Anders Hejlsberg, le concepteur de TypeScript, a annoncé il y a deux semaines que son équipe travaillait à l'implémentation des unions de types : [lien 1](#). L'union de types est un concept assez simple

à comprendre, mais étrangement peu répandu dans les langages typés. Voici la traduction de la spécification #805 rédigée par Ryan Cavanaugh dans sa version originale : [lien 2](#).

## 2 L'opérateur | sur les types

Ceci est un résumé des spécifications imaginées par Anders Hejlsberg.

le type *any*, perdant ainsi la sécurité du typage.

De même, la configuration du service HTTP d'AngularJS ([lien 3](#)) possède des propriétés de type *boolean*, *Cache*, *number* ou encore *Promise*.

### 2.1 Cas d'utilisation

Beaucoup de bibliothèques JavaScript acceptent des valeurs de plus d'un seul type. Par exemple, la propriété AJAX *jsonp* avec jQuery peut être soit *false* (i.e. de type *boolean*) ou une chaîne de caractères (type *string*). Les fichiers de définition TypeScript (*.d.ts*) doivent représenter cette propriété avec

### 2.2 Solutions de contournement actuelles

Cette lacune peut souvent être contournée avec des surcharges de fonction, mais il n'y a pas d'équivalent pour les propriétés des objets, les contraintes sur les types, ou d'autres rôles concernant les types.

## 3 Présentation

### 3.1 Syntaxe

Le nouvel opérateur |, lorsqu'il est utilisé pour séparer deux types, produit une union de types représentant une valeur qui est de l'un des types en entrée.

Exemple :

```
1 interface Settings {
2     foo: number|string;
3 }
4 function setSetting(s: Settings) { /*
5     ... */ }
6 setSettings({ foo: 42 }); // OK
7 setSettings({ foo: '100' }); // OK
8 setSettings({ foo: false }); // Error,
9     false is not assignable to number|
10    string
```

Plusieurs types peuvent être combinés de cette façon :

```
1 function process(n: string|HTMLElement|
2     JQuery) { /* ... */ }
3 process('foo'); // OK
4 process($('div')); // OK
5 process(42); // Error
```

N'importe quel type est un opérande valide pour l'opérateur |. Voici quelques exemples et comment ils seraient analysés :

```
1 var x: number|string[]; // x is a number
2     or a string[]
3 var y: number[]|string[]; // y is a
4     number[] or a string[]
5 var z: Array<number|string>; // z is an
6     array of number|string
```

```

4 var t: F|typeof G|H; // t is an F,
  typeof G, or H
5 var u: () => string|number; // u is a
  function that returns a string|
  number
6 var v: { (): string; }|number; // v is a
  function that returns a string, or
  a number

```

Notez que les parenthèses n'étant pas nécessaires

## 4 Sémantique

### 4.1 Notions de base

Quelques règles simples :

- identité :  $A|A$  est équivalent à  $A$ ;
- commutativité :  $A|B$  est équivalent à  $B|A$ ;
- associativité :  $(A|B)|C$  est équivalent à  $A|(B|C)$ ;
- effacement du sous-type :  $A|B$  est équivalent à  $A$  si  $B$  est un sous-type de  $A$ .

### 4.2 Propriétés (attributs)

Le type  $A|B$  possède une propriété  $P$  de type  $X|Y$  si  $A$  possède une propriété  $P$  de type  $X$  et  $B$  possède une propriété  $P$  de type  $Y$ . Ces propriétés doivent soit être à la fois publiques, soit provenir du même site de déclaration (tel que spécifié dans les règles pour *private/protected*). Si l'une des propriétés est facultative, la propriété qui en résulte est également facultative.

## 5 Appel et construction de signatures

Le type  $A|B$  a une signature d'appel  $F$  si  $A$  a une signature d'appel  $F$  et  $B$  a une signature d'appel  $F$ .

Exemple :

```

1 var t: string|boolean = /* ... */;
2 console.log(t.toString()); // OK (both
  string and boolean have a toString
  method)

```

## 6 Réductibilité (assignability) et sous-typage

Nous décrivons ici la réductibilité ; le sous-typage est la même chose, sauf que « *est réductible à* » est remplacé par « *est un sous-type de* ».

Le type  $S$  est réductible au type  $T1|T2$  si  $S$  est réductible à  $T1$  ou si  $S$  est réductible à  $T2$ .

Exemple :

```

1 var x: string|number;
2 x = 'hello'; // OK, can assign a string
  to string|number
3 x = 42; // OK
4 x = { }; // Error, { } is not assignable
  to string or assignable to number

```

pour lever l'ambiguïté, elles ne sont pas acceptées.

### 3.2 Interprétation

La signification de  $A|B$  est un type qui est soit un  $A$ , soit un  $B$ . En particulier, c'est différent d'un type qui combinerait tous les membres de  $A$  et de  $B$ . Nous examinerons cela dans des exemples plus loin.

Exemple :

```

1 interface Car {
2   weight: number;
3   gears: number;
4   type: string;
5 }
6 interface Bicycle {
7   weight: number;
8   gears: boolean;
9   size: string;
10 }
11 var transport: Car|Bicycle = /* ... */;
12 var w: number = transport.weight; // OK
13 var g = transport.gears; // OK, g is of
  type number|boolean
14
15 console.log(transport.type); // Error,
  transport does not have 'type'
  property
16 console.log((<Car>transport).type); //
  OK

```

La même règle est appliquée pour construire signatures.

### 5.1 Signatures d'indice

Le type  $A|B$  a une signature d'indice  $[x : \textit{number}] : T$  ou  $[x : \textit{string}] : T$  si les deux  $A$  et  $B$  ont une signature d'indice de ce type.

Le type  $S1|S2$  est réductible au type  $T$  si les deux  $S1$  et  $S2$  sont réductibles à  $T$ .

```

1 var x: string|number = /* ... */;
2 var y: string = x; // Error, number is
  not assignable to string
3 var z: number = x; // Error, string is
  not assignable to number

```

En combinant les règles, le type  $S1|S2$  est réductible au type  $T1|T2$  si  $S1$  est réductible à  $T1$  ou  $T2$  et  $S2$  est réductible à  $T1$  ou  $T2$ . Plus généralement, tous les types sur la partie droite de la réduction

doivent être réduits à au moins un type sur la partie gauche.

Exemple :

```
1 var x: string|number;
2 var y: string|number|boolean;
```

## 7 Meilleur type commun

L'algorithme actuel du meilleur type commun (c.f. spécifications section 3.10) est seulement capable de produire un type déjà existant parmi les candidats, ou le type `.` Par exemple, le tableau `[1, 2, "hello"]` est de type `[]`. Avec la possibilité de représenter les unions de types, nous pouvons changer l'algorithme du meilleur type commun pour produire une union de types lorsqu'on est en présence d'un ensemble de candidats sans supertype.

Exemple :

```
1 class Animal { run(); }
2 class Dog extends Animal { woof(); }
3 class Cat extends Animal { meow(); }
4 class Rhino extends Animal { charge(); }
5 var x = [new Dog(), new Cat()];
6 // Current behavior: x is of type {}[]
7 // Proposed: x is of type Array<Dog|Cat>
```

Notez que dans ce cas, le type `Dog|Cat` est structurellement équivalent à `Animal` par rapport à ses membres, mais il serait une erreur d'essayer d'attribuer un `Rhino` à `x[0]`, car `Rhino` n'est pas réductible à `Cat` ou `Dog`.

Le meilleur type commun est utilisé pour plusieurs inférences réalisées par le langage. Dans les cas

```
3 x = y; // Error, boolean is not
    assignable to string or number
4 y = x; // OK (both string and number are
    assignable to string|number)
```

- $x \parallel y$ ,
- $z ? x : y$ ,
- $z ? x : y$  et
- $[x, y]$ ,

le type résultant sera  $X | Y$  (où  $X$  est le type de  $x$  et  $Y$  est le type de  $y$ ). Pour l'instruction `return` dans une fonction et pour l'inférence du type générique, nous allons exiger l'existence d'un supertype entre les candidats.

Exemple :

```
1 // Error, no best common type among '
    string' and 'number'
2 function fn() {
3   if(Math.random() > 0.5) {
4     return 'hello';
5   } else {
6     return 42;
7   }
8 }
9 // OK with type annotation
10 function fn(): string|number {
11   /* ... same as above ... */
12 }
```

## 8 Prochaines étapes possibles

### 8.1 Combinaison des membres de types

D'autres scénarios nécessitent un type construit à partir de  $A$  et  $B$  ayant tous ses membres présents dans l'un ou l'autre des deux types, mais pas dans les deux. Au lieu d'ajouter une nouvelle syntaxe de type, nous pouvons représenter cela facilement en supprimant la restriction qui fait que les clauses `extends` peuvent ne pas référencer les paramètres de type de leur déclaration.

Exemple :

```
1 interface HasFoo<T> extends T {
2   foo: string;
3 }
4 interface Point {
5   x: number;
6   y: number;
7 }
8 var p: HasFoo<Point> = /* ... */;
9 console.log(p.foo); // OK
10 console.log(p.x.toString()); // OK
```

### 8.2 Signification locale des unions de types

Pour les unions de types où un opérande est une primitive, nous avons pu détecter certains schémas syntaxiques et ajuster le type d'un identifiant dans les blocs conditionnels.

Exemple :

```
1 var p: string|Point = /* ... */;
2 if(typeof p === 'string') {
3   console.log(p); // OK, 'p' has type
    string in this block
4 } else {
5   console.log(p.x.toString()); // OK,
    'p' has type Point in this block
6 }
```

Cela pourrait également s'étendre à des vérifications d'appartenance :

```
1 interface Animal { run(); }
2 interface Dog extends Animal { woof(); }
3 interface Cat extends Animal { meow(); }
4 var x: Cat|Dog = /* ... */;
5 if(x.woof) {
6   // x is 'Dog' here
7 }
```

```
8 if(typeof x.meow !== 'undefined') {
9   // x is 'Cat' here
10 }
```

Retrouvez l'article de l'équipe TypeScript traduit par Yahiko en ligne : [lien 4](#)

# Introduction au langage TypeScript - Le JavaScript++ ?

Cet article est une introduction au langage TypeScript, alternative intéressante au JavaScript de base.

## 1 Introduction

TypeScript ([lien 5](#)) est un langage très récent (février 2012) qui a été conçu par Anders Hejlsberg, également concepteur du langage C#. Le but premier de TypeScript est de rendre plus facile et plus fiable l'écriture de code en JavaScript pour des applications de grande ampleur.

Cet article présentera de façon succincte les principaux ajouts de TypeScript par rapport à JavaScript comme le typage, les classes, et la modularité,

sachant que pour ceux connaissant déjà JavaScript, l'apprentissage de TypeScript ne devrait être qu'une simple formalité.

Puis, après la brève mention des principaux outils de développement utilisables avec TypeScript, nous comparerons l'approche de TypeScript avec deux autres langages similaires, CoffeeScript et Dart, et tenterons de montrer en quoi l'approche de TypeScript semble la plus viable à l'avenir.

## 2 Typage

### 2.1 Variables

Le principal apport du langage TypeScript, celui qui justifie le nom même du langage, est la possibilité d'associer, facultativement, un type à une donnée.

```
1 var pi: number;
2 var message: string;
3 var flag: boolean;
4 var joker: any;
```

Dans l'exemple ci-dessus, quatre variables sont déclarées sans être initialisées à l'aide d'un type dont la signification est explicite.

- La variable *pi* a pour type *number*, un nombre entier ou flottant.
- La variable *message* a pour type *string*, une chaîne de caractères.
- La variable *flag* a pour type *boolean*, un booléen qui peut prendre la valeur *true* ou *false*.
- La variable *joker* a pour type *any*, qui est le type par défaut qu'attribue TypeScript à une variable globale s'il ne parvient pas à déterminer son type lors de sa déclaration.

Bien entendu, il est possible d'initialiser une variable au moment de sa déclaration comme on peut le voir ci-dessous :

```
1 var pi = 3.14; // number
2 var message = "Bonjour !"; // string
3 var flag = true; // boolean
4 var joker = null; // any
```

Lors de la première initialisation d'une variable, TypeScript en infère automatiquement le type sans qu'il soit nécessaire de le mentionner explicitement.

Ainsi, TypeScript, contrairement à JavaScript, peut être considéré comme un langage à typage statique.

### 2.2 Fonctions

Il est courant qu'une fonction renvoie un certain résultat. Le langage TypeScript permet de préciser le type du résultat attendu lors de la déclaration de la fonction.

Par défaut et en l'absence d'instruction *return*, le type du résultat d'une fonction est *void*, c'est-à-dire aucun résultat.

```
1 function triple(n: number): number {
2   return 3 * n;
3 }
```

La fonction *triple* ci-dessus est déclarée comme prenant un paramètre de type *number* et renvoyant une valeur de type *number*.

### 3 Classe

La notion de classe introduite dans TypeScript anticipe la prochaine évolution de JavaScript (ECMAScript 6).

```

1 class Animal {
2   name: string;
3
4   constructor(name: string) {
5     this.name = name;
6   }
7
8   shout(): string {
9     return "...";
10  }
11 }

```

Comme on peut le voir dans l'exemple ci-dessus, une classe *Animal* y est définie d'une façon proche de la plupart des langages orientés objet.

La classe *Animal* possède ici un attribut (*name*), elle définit un constructeur (*constructor*) et une méthode (*shout*). Son instantiation se fait à l'aide de l'opérateur *new* comme ceci :

```
1 var animal = new Animal("pokemon");
```

De pair avec la notion de classe, TypeScript implémente la notion d'héritage simple par l'utilisation du mot-clé *extends*.

L'extension de la classe *Animal* de l'exemple précédent pourrait se faire ainsi :

```

1 class Lion extends Animal {
2   sex: string;

```

### 4 Interface

Une interface peut être vue tout d'abord comme une sorte de contrat minimum que doit respecter une structure de données en termes d'attributs et de méthodes. Cette structure de données peut être un objet ... ou une classe.

```

1 interface I1 {
2   a: number;
3 }
4
5 interface I2 {
6   b: string;
7 }

```

L'interface *I1* ci-dessus indique que la structure de données implémentant *I1* doit avoir à minima un attribut nommé *a* de type *number*. L'interface *I2* indique que la structure de données l'implémentant doit avoir à minima un attribut nommé *b* de type *string*.

Une interface est considérée comme un type à part entière, ce qui signifie qu'il est possible de définir une fonction qui prendrait un paramètre de type *I1* comme l'exemple ci-dessous :

```

1 function fct(x: I1) {
2   alert(x.a);
3 }

```

```

3
4   constructor(name: string, sex: string)
5   {
6     super(name);
7     this.sex = sex;
8   }
9
10  shout(): string {
11    return "Roocaaarrr!";
12  }
13 }

```

Cette nouvelle classe *Lion* ajoute un nouvel attribut *sex* à la classe *Animal* et redéfinit la méthode *shout*.

Puisque toutes les classes définies dans TypeScript sont considérées comme de nouveaux types, la classe *Lion* est du type *Lion*, et en vertu de l'héritage est aussi du type *Animal*.

En remarque, malgré cet apport orienté objet à la syntaxe initiale de JavaScript, il faut avoir conscience que les limitations intrinsèques de JavaScript se reflètent également dans TypeScript. Par exemple, la notion d'attribut privé (*private*) d'une classe qui existe dans la plupart des langages orientés objet, bien que syntaxiquement présente dans TypeScript, n'est pas véritablement restrictive dans la mesure où un attribut privé pourra malgré tout être utilisé en dehors de sa classe avec une approche dynamique.

Cette fonction *fct* pouvant être appelée en passant en paramètre un objet ayant au moins un attribut nommé *a* :

```
1 fct({ a: 20, z: -1 });
```

Un autre intérêt d'une interface, c'est qu'elle peut être considérée comme une classe n'ayant que des méthodes abstraites (ou pas de méthode du tout). Cela permet dans un langage comme TypeScript ne gérant que l'héritage simple, de se rapprocher du concept de l'héritage multiple, une classe pouvant implémenter (*implements*) plusieurs interfaces comme le montre l'exemple ci-dessous :

```

1 class C implements I1, I2 {
2   a: number;
3   b: string;
4
5   constructor(a: number, b: string) {
6     this.a = a;
7     this.b = b;
8   }
9 }

```

Une instance de cette classe *C* pourrait tout à fait être passée en paramètre de notre fonction *fct* définie plus haut.



```
1 var c = new C(15, "bonjour");
2 fct(c);
```

Enfin, comme nous venons de le voir, une interface pouvant être considérée comme une classe abstraite, il est possible de faire hériter une interface d'une autre à l'aide du mot-clé *extends* :

## 5 Fonction anonyme fléchée

Un autre apport notable de TypeScript à JavaScript est sa nouvelle manière de définir une fonction anonyme qui peut être dénommée ici notation « fléchée » (*arrow function*) en raison de l'utilisation du symbole `=>`.

```
1 function alertMsg(msg: string) {
2   this.msg = msg;
3   this.timer = setInterval(
4     () => {
5       alert(this.msg);
6     },
7     500);
8 }
```

La fonction anonyme fléchée est définie dans cet exemple en premier paramètre de l'appel à *setInterval*. À gauche du symbole `=>` doit être mentionnée la liste de paramètres entre parenthèses, et à droite du symbole `=>`, le corps de la fonction anonyme entre accolades.

L'avantage par rapport à l'actuelle norme de JavaScript (ECMAScript 5), est que la notation fléchée ne change pas la valeur de contexte du mot-clé *this*

## 6 Modularité

La modularité en TypeScript est un sujet qui peut rapidement devenir complexe et nécessiterait un article en soi. Dans ce chapitre, nous nous contenterons de n'évoquer que les principales techniques.

### 6.1 Référencement de fichiers

TypeScript introduit de nouvelles techniques de modularisation devenues nécessaires par le typage statique et par l'héritage de classes. Parmi ces techniques de modularisation, le référencement d'un fichier TypeScript externe est peut-être la plus simple.

Elle consiste à inclure de façon statique en tout début de fichier, un autre fichier source TypeScript contenant des déclarations nécessaires au compilateur afin qu'il en déduise entre autres les types et les héritages.

```
1 /// 
```

À noter que le référencement ressemble un peu aux `#include` des langages C/C++.

C'est cette approche qui est en général adoptée pour utiliser des bibliothèques JavaScript déjà existantes via les fichiers de déclaration *.d.ts*.

```
1 interface I3 extends I2 {
2   c: boolean;
3 }
```

Dans l'exemple ci-dessus, la nouvelle interface *I3* dérive de l'interface *I2* définie plus haut en ajoutant un attribut *c* de type *boolean*.

à l'intérieur de la fonction anonyme. Cela évite donc l'utilisation d'une variable intermédiaire servant à propager la valeur de *this* à l'intérieur de la fonction anonyme. Par exemple, la ligne de code précédente aurait pu être transcrite ainsi en JavaScript actuel :

```
1 function alertMsg(msg: string) {
2   this.msg = msg;
3   var _this = this;
4   this.timer = setInterval(
5     function () {
6       alert(_this.msg);
7     },
8     500);
9 }
```

Cette nouvelle notation plus concise sera utile en particulier pour les fonctions *callback* appelées de façon asynchrone.

Il convient aussi de noter que la norme ECMAScript 6, si elle est approuvée, reprendra cette notation ; TypeScript permettant ainsi de se familiariser en avance avec ces fonctions anonymes fléchées.

Sur certains environnements intégrés de développement comme *Visual Studio*, le référencement peut se faire implicitement, sans avoir besoin de mentionner le fichier contenant les dépendances de déclarations, même si un référencement explicite peut faciliter la documentation des dépendances entre fichiers sources.

Ce référencement permet très simplement de bénéficier à l'intérieur du fichier source appelant, partageant le même espace de nommage, des déclarations de variables, de fonctions ou de classes présentes dans le fichier référencé.

L'utilisation du référencement de fichiers se fait généralement pour les applications côté client où dans la page HTML il sera nécessaire soit de référencer via la balise `<script>` l'ensemble des fichiers JavaScript générés (c.f. *index1.htm*), soit de référencer le fichier issu de la concaténation éventuellement minifiée (*.min.js*) de ces fichiers JavaScript (c.f. *index2.htm*).

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <script src="module.js"></script>
```

```

5 <script src="source.js"></script>
6 </head>
7 <body>
8   ...
9 </body>
10 </html>

```



Dans l'exemple ci-dessus, on suppose que les fichiers *module.js* et *source.js* sont le résultat de la compilation en JavaScript des fichiers *module.ts* et *source.ts*.

```

1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <script src="source-module.min.js"></script>
5 </head>
6 <body>
7   ...
8 </body>
9 </html>

```



Dans l'exemple ci-dessus, on suppose que le fichier *source-module.min.js* est la concaténation du fichier *module.js* suivi du fichier *source.js*.

Par souci de performance, la seconde solution consistant à fusionner les fichiers JavaScript résultants est en général privilégiée. C'est pourquoi les fichiers reliés par référencement (dans l'exemple *module.ts*) au fichier source principal (dans l'exemple *source.ts*) peuvent être qualifiés de modules internes, par opposition aux modules externes reliés par une autre méthode et que nous verrons un peu plus loin.

L'inconvénient de cette approche est que cela peut engendrer des conflits de nommage entre deux variables globales, fonctions ou classes pouvant porter la même appellation. Pour remédier à ce problème, TypeScript propose la notion de module explicite.

## 6.2 Module explicite

En TypeScript chaque fichier source est implicitement un module.

Il est possible de déclarer explicitement un module, ce qui basiquement créera un espace de nommage permettant de spécifier les variables, fonctions, interfaces ou classes pouvant être importées dans un autre fichier, par exemple via référencement. Un module explicite est introduit par le mot-clé *module*.

```

1 module M {
2   var temp = "bonjour";
3   export var id = 0;
4   export function fct() { ... }
5   export interface I {
6     a: number;
7   }
8   export class C implements I {
9     a: number;

```

```

10 }
11 }

```

Dans l'exemple ci-dessus, le module *M* exporte, c'est-à-dire rend accessible, les éléments suivants :

- la variable *id* ;
- la fonction *fct* ;
- l'interface *I* ;
- et la classe *C*.

La variable *temp* est quant à elle interne à ce module *M* et ne pourra pas être utilisée en dehors de ce module.

Par exemple, dans ce même fichier source, l'instruction suivante utilisant la variable *id* est valide :

```
1 alert(M.id);
```

alors que l'instruction ci-dessous utilisant la variable *temp* n'est pas valide :

```
1 alert(M.temp); // erreur
```

## 6.3 Importation de modules externes

Alors que précédemment la responsabilité de l'espace de nommage était au niveau du module appelé, forçant l'utilisateur à réutiliser l'espace de nommage du module concerné, il est possible de transférer la responsabilité de l'espace de nommage au niveau du fichier source appelant à l'aide des mot-clés *import* et *require*.

Reprenons l'exemple de la section précédente, mais cette fois sans créer d'espace de nommage via le mot-clé *module* :

```

1 export var id = 0;
2 export function fct() { ... }
3 export interface I {
4   a: number;
5 }
6 export class C implements I {
7   a: number;
8 }

```

Ainsi défini, ce fichier *module2.ts* peut être importé dans un autre fichier, qu'on supposera pour l'exemple présent dans le même répertoire que *module2.ts*, de la manière suivante :

```

1 import m = require("./module2");
2
3 m.id++;

```

Ci-dessus, le symbole *m* ainsi défini via le mot-clé *import* est un nouvel espace de nommage englobant les éléments exportés du fichier *module2.ts*, sans avoir recours au référencement de fichiers via la syntaxe */// <reference>*.

Cette technique de modularisation est surtout pratiquée pour des applications côté serveur (e.g. Node.js) où les différents fichiers n'ont pas vocation à être combinés en un seul fichier final JavaScript. C'est la raison pour laquelle on peut parler de modules externes dans ce cas.



## 7 Typage générique

Le typage générique ajoute un niveau d'abstraction en rendant les types paramétrables, que ce soit dans une fonction, une classe ou une interface, sachant qu'au moment de l'appel effectif à cette fonction ou à cette classe, le type devra être explicitement défini.

Le typage générique est surtout utile en TypeScript lorsqu'il s'agit d'assurer la cohérence du typage entre différents éléments. Par exemple, si on considère un tableau de nombres, on pourrait souhaiter disposer d'une fonction de concaténation assurant de concaténer ce tableau de nombres avec un autre tableau de nombres et non avec un tableau de chaînes de caractères ou d'autres choses. Par contre, on pourrait souhaiter que cette même fonction de concaténation puisse s'appliquer sur deux tableaux de chaînes de caractères ou autres, sans avoir besoin de réécrire une nouvelle fonction pour chaque type traité. C'est à ce moment-là qu'intervient le typage générique.

```

1 function concatenate<T>(a1: T[], a2: T
  []): T[] {
2     return a1.concat(a2);
3 }
4
5 resultNumbers = concatenate<number>([1,
```

```

  2], [3, 4]); // [1, 2, 3, 4]
6 resultStrings = concatenate<string>(["a",
  "b"], ["c", "d"]); // ["a", "b", "
  c", "d"]
7 resultError1 = concatenate<number>([1, 2
  ], ["a", "b"]); // erreur
8 resultError2 = concatenate<string>([1, 2
  ], ["a", "b"]); // erreur
9 resultAny = concatenate<any>([1, 2], ["a",
  "b"]); // [1, 2, "a", "b"]
```

La syntaxe du typage générique est tout ce qu'il y a de classique pour les habitués de C++, C# ou de Java. Le type générique (dans l'exemple *T*) est déclaré entre chevrons <>, puis est utilisé comme un type ordinaire.

L'exemple montre la concaténation de deux tableaux de nombres (*resultNumbers*) et de deux tableaux de chaînes de caractères (*resultStrings*), et montre aussi qu'il n'est pas possible de concaténer un tableau de nombres avec un tableau de chaînes de caractères (*resultError1* et *resultError2*) sous peine de lever une erreur.

Le typage générique est donc différent du type *any* puisqu'une fonction de concaténation entre deux tableaux de type *any* n'aurait pas levé d'erreur au niveau de la compilation comme l'indique le quatrième résultat de l'exemple (*resultAny*).

## 8 Outils

Bien que TypeScript soit d'abord un compilateur installable via Node.js ([lien 6](#)), pour être exploité à son plein potentiel, il est recommandé qu'il soit associé à un environnement intégré de développement comme Visual Studio ([lien 7](#)), WebStorm ([lien 8](#)) ou éventuellement Eclipse ([lien 9](#)).

Pour tester de petits morceaux de codes (snip-

pets), il peut être intéressant d'utiliser le site TypeScript.io ([lien 10](#)) fonctionnant de façon similaire à JsFiddle ([lien 11](#)).

Aussi, la page de test ([lien 12](#)) mise à disposition par Microsoft peut également être utile pour voir immédiatement la transcription en JavaScript d'un code TypeScript.

## 9 TypeScript et les autres

Après cet aperçu concernant le « sucre syntaxique » de TypeScript, intéressons-nous au modèle de développement de ce langage comparativement à ses principaux concurrents que sont CoffeeScript (décembre 2009) et Dart (2011).

### 9.1 Maturité

On constate tout d'abord que TypeScript est le plus jeune des trois, n'étant apparu qu'en 2012. Ce retard de maturité est encore visible dans la mesure où la version de production 1.0 n'a été disponible que cette année 2014. Cependant, force est de constater qu'à l'instar de Google avec Dart, Microsoft et son concepteur restent très impliqués dans l'évolution de TypeScript, ce qui ne peut que rassurer ses utilisateurs.

Concernant TypeScript, il est raisonnable de tabler sur deux à trois évolutions majeures par an.

### 9.2 Code source

Tout comme CoffeeScript et Dart, le code source du compilateur TypeScript est disponible en *Open source* (Licence Apache 2), sur le compte GitHub de Microsoft : [lien 13](#). Ce critère n'est donc pas discriminant pour évaluer la pertinence relative de ce langage sur les deux autres.

### 9.3 Correspondance JavaScript

La conversion de TypeScript vers le JavaScript est assez directe et le sera encore davantage avec

l'officialisation de la norme ECMAScript 6 puisque TypeScript est un sur-ensemble de JavaScript. Il est donc très aisé d'identifier le morceau de code TypeScript ayant produit un morceau de code JavaScript. Cette forte correspondance (voulue) entre TypeScript et JavaScript permet à TypeScript de bénéficier sans trop d'efforts aux innombrables bibliothèques, API ou frameworks écrits en JavaScript. C'est évidemment un gros avantage.

Ce n'est pas forcément le cas des deux autres langages. La syntaxe de CoffeeScript inspirée du langage Haskell peut parfois être absconse et en tous les cas significativement différente de JavaScript pouvant rendre le débogage, l'identification entre code source et le code traduit et l'importation de codes en JavaScript natifs non triviaux. Et cela est encore plus vrai pour Dart qui emploie une syntaxe différente à celle de JavaScript en s'inspirant davantage de Java et du C++. D'autant plus que les optimisations réalisées par Dart lors de la conversion en code JavaScript peuvent significativement changer la structure du code et l'ordre des instructions.

En fin de document, on trouvera en annexe le résultat de la conversion en JavaScript d'un programme simple rédigé en trois versions : CoffeeScript, Dart et TypeScript. Le lecteur se convaincra facilement que TypeScript est syntaxiquement le langage le plus proche de JavaScript, le code source étant très similaire au code cible, tandis que dans le cas de Dart avec son code cible interminable, on peut se poser la question de son lien avec JavaScript.

#### 9.4 Portabilité

Les trois langages permettent la conversion (ou transcompilation) en JavaScript et donc l'accès en théorie à tous les navigateurs du marché.

Cependant, pour bénéficier pleinement des performances promises par Dart, il est nécessaire que le navigateur dispose d'une machine virtuelle DartVM, ce qui n'est le cas actuellement que pour Chrome, évidemment. Dans la mesure où il est peu probable que Microsoft et Apple implémentent une DartVM dans Internet Explorer et dans Safari, il est raisonnable de considérer que ce qui est souvent vendu comme l'avantage décisif de Dart, ne soit finalement réservé qu'à une fraction des navigateurs. Par conséquent, Dart, à terme, devrait vraisemblablement rester marginal. Un peu comme Google Web Toolkit (2006) dont on retrouve beaucoup de caractéristiques avec Dart, et qui n'a pas vraiment eu de succès comme on peut le constater aujourd'hui.

C'est pourquoi JavaScript, en tant que langage

natif de tous les navigateurs, devrait rester le seul standard pendant encore de nombreuses années.

Et l'approche de TypeScript se voulant la plus respectueuse des normes de JavaScript a de grandes chances d'être la bonne.

#### 9.5 Popularité

Bien que les classements de popularité soient toujours sujets à caution, et ne mentionnons même pas l'indice très douteux de Tiobe, c'est souvent sur ce point que se décide l'adoption d'un langage, ce sursisme entraînant un effet boule de neige dont chacun est libre de penser ce qu'il voudra. Mais c'est un fait et cela a son importance.

Si nous nous basons sur la plate-forme d'hébergement de codes source GitHub (lien 14) qui est très prisée des développeurs Web comme l'indique la position dominante de JavaScript sur les autres langages, nous devrions avoir une tendance objective relativement proche des usages dans ce domaine.

Langage	2012-2013	2013-2014	+/-
JavaScript	531 482	702 596	32 %
CoffeeScript	26 362	42 094	60 %
Dart	3 120	3 393	9 %
TypeScript	2 150	6 558	205 %

*Évolution* (Réalisée à partir d'une requête Google BigQuery sur le site githubarchive (lien 15)) annuelle (d'octobre à septembre) de la création de repositories sur GitHub.

On peut constater que la croissance de Dart (9 %) reste assez stable et en deçà de ses concurrents, y compris JavaScript lui-même (32 %). CoffeeScript continue quant à lui de connaître une belle dynamique avec une croissance des créations de +60 %. TypeScript qui comptabilisait en 2012-2013 moins de créations de *repositories* que Dart, vient de le dépasser en 2013-2014 avec un nombre de créations deux fois plus fort et une croissance annuelle qui explose à +205 %.

Ceci est un indicateur laissant à penser que TypeScript a un grand potentiel d'adoption parmi la communauté des développeurs Web malgré sa jeunesse. On peut signaler à titre d'exemple que le framework graphique *babylon.js* (lien 16) a annoncé récemment la migration de son code de JavaScript vers TypeScript : lien 17. Gageons qu'avec la récente mise à disposition d'une version de production et donc stable du compilateur TypeScript, d'autres migrations d'ampleur vers ce langage voient le jour dans les mois qui viennent.

## 9.6 Récapitulatif

	CoffeeScript	Dart	TypeScript
<b>Maturité</b>	5 ans	4 ans	3 ans
<b>Code source</b>	<i>Open source</i>	<i>Open source</i>	<i>Open source</i>
<b>Correspondance</b>	Moyenne	Faible	Forte
<b>Portabilité</b>	JavaScript	JavaScript sans DartVM	JavaScript
<b>Popularité</b>	Restreinte, forte croissance	Confidentielle, faible croissance	Confidentielle, très forte croissance

## 10 Conclusion

Cette présentation est évidemment loin d'être complète. Des mécanismes comme les mixins n'ont pas été abordés tout comme les possibilités offertes avec les fonctions concernant les paramètres et la surcharge. Pour se familiariser avec TypeScript, il peut être utile d'utiliser la page de test ([lien 18](#)) afin de voir instantanément la conversion d'un code TypeScript en un code JavaScript. Sinon, le guide d'utilisation ([lien 19](#)) et la spécification ([lien 20](#)), malheureusement ni l'un ni l'autre n'étant traduit en français, restent indispensables pour approfondir le sujet.

En dépit de sa courte existence, TypeScript est un langage en pleine croissance tant sur le plan de ses spécifications que sur le plan de sa popularité. Il

n'est pas parfait du fait de sa volonté de coller au plus près des normes de JavaScript, mais l'évolution de ces normes pour enrichir la syntaxe de JavaScript ainsi que des approches parallèles pour améliorer les performances comme `asm.js` ([lien 21](#)) permettent de penser que TypeScript pourrait à terme au mieux devenir lui-même un standard, au pire, inspirer les futures normes ECMAScript.

Toute proportion gardée, c'est une démarche similaire qu'avait eue à son époque Bjarne Stroustrup en concevant le C++ en tant que sur-ensemble du langage C. Et le succès jamais démenti du C++ depuis plusieurs décennies laisse à penser que TypeScript pourrait suivre le même chemin et devenir une sorte de JavaScript++.

## 11 Annexe : conversion en JavaScript

Les trois codes source proviennent du compte GitHub `darthapo` : [lien 22](#). Les codes cibles ont été réactualisés avec les derniers compilateurs en date.

### 11.1 CoffeeScript 1.8.0

```

1 class Simple
2
3   constructor: (@name = "default") ->
4
5   greet: (who) ->
6     "Greetings #{ who }, I'm #{ @name }!"
7
8   @main: ->
9     s = new Simple "Flynn"
10    console.log s.greet "Program"
11
12 Simple.main();

```

```

1 (function() {
2   var Simple;
3
4   Simple = (function() {
5     function Simple(name) {
6       this.name = name != null ? name :
          "default";

```

```

7     }
8
9     Simple.prototype.greet = function(
10      who) {
11       return "Greetings " + who + ", I'm
12         " + this.name + "!";
13     };
14
15     Simple.main = function() {
16       var s;
17       s = new Simple("Flynn");
18       return console.log(s.greet("
19         Program"));
20     };
21
22     return Simple;
23   })();
24
25   Simple.main();
26 })().call(this);

```

### 11.2 Dart 1.7.2

```

1 class Simple {
2   String name;
3
4

```

```

5   Simple(this.name);
6
7   greet(who) => "Greetings $who, I'm
      $name!";
8 }
9
10 main() {
11   var s = new Simple('Flynn');
12
13   print( s.greet('Program'));
14 }

```

Voir le code en ligne : [lien 23](#)

### 11.3 TypeScript 1.1

```

1 class Simple {
2
3   constructor(public name: string) {}
4
5   greet(who: string) {
6     return "Greetings " + who + ", I'm " +
7       this.name + "!";
8   }
9
10  static main() {
11    var s = new Simple('Flynn');

```

```

11     console.log(s.greet("Program"));
12   }
13 }
14 }
15
16 Simple.main()

```

```

1 var Simple = (function () {
2   function Simple(name) {
3     this.name = name;
4   }
5   Simple.prototype.greet = function (
6     who) {
7     return "Greetings " + who + ", I
8       'm " + this.name + "!";
9   };
10  Simple.main = function () {
11    var s = new Simple('Flynn');
12    console.log(s.greet("Program"));
13  };
14  return Simple;
15 })();
16 Simple.main();

```

Retrouvez l'article de **yahiko** en ligne : [lien 24](#)

# Génération aléatoire de noms - De Elite aux chaînes de Markov

Cet article s'intéresse aux techniques de génération aléatoire de noms et à une d'entre elles en particulier : les chaînes de Markov.

## 1 Introduction

Autant pour un être humain, il est facile d'inventer des mots nouveaux, autant pour un ordinateur, ce n'est pas si simple que cela en à l'air. Il ne s'agit évidemment pas de fabriquer des suites de caractères purement aléatoires, qui seraient imprononçables, mais bien de produire des noms (de personnes, de lieux, d'espèces, etc.) ayant une certaine vraisemblance.

C'est un besoin assez connu dans l'univers des jeux de rôle où la multitude de personnages et de lieux fait qu'une génération procédurale peut soulager grandement la tâche du Maître du Jeu (dans le cas des jeux de rôle papier) ou des concepteurs (dans le cas des jeux de rôle informatiques).

Nous essayerons donc dans cette présentation de trouver une solution à cette génération aléatoire (mais vraisemblable) de noms.

Puisque nous aborderons le concept de **chaînes de Markov** qui ont des applications diverses, cet article peut éventuellement être utile aux personnes souhaitant une introduction à ces chaînes de Markov, indépendamment de la question de la génération aléatoire de noms.

Le langage utilisé pour illustrer les algorithmes sera TypeScript, un surensemble de JavaScript avec typage et héritage. Une présentation de ce langage est disponible **ici** : [lien 25](#).

## 2 Elite



Version originale du jeu Elite sur BBC Micro

*Elite* est un jeu célèbre des années 1980 qui a posé les bases des mondes ouverts et de la génération procédurale. Parmi les nombreux éléments générés à chaque nouvelle partie, se trouve le nom des étoiles.

La génération du nom des étoiles dans *Elite* est très simple ce qui ne l'empêche pas d'être efficace. L'algorithme va piocher aléatoirement dans une liste prédéfinie, quatre paires de lettres (consonne puis voyelle ou inversement) formant ainsi quatre syllabes qui formeront le nom final.

Voici ce que pourrait donner l'implémentation en TypeScript :

```
1 function randNameElite(): string {
2   var pairs = "..lexegezacebiso"
3             "usesarmaindirea."
4             "eratenberalaveti"
5             "edorquanteisrion";
6
7   var pair1 = 2 * Math.floor(Math.random() * (pairs.length / 2));
8   var pair2 = 2 * Math.floor(Math.random() * (pairs.length / 2));
9   var pair3 = 2 * Math.floor(Math.random() * (pairs.length / 2));
10  var pair4 = 2 * Math.floor(Math.random() * (pairs.length / 2));
11
```

```
12   var name = "";
13   name += pairs.substr(pair1, 2);
14   name += pairs.substr(pair2, 2);
15   name += pairs.substr(pair3, 2);
16   name += pairs.substr(pair4, 2);
17   name = name.replace(/[/./]/g, "");
18
19   return name;
20 } // randNameElite
```



Pour les plus curieux, le concepteur du jeu, Ian Bell, a mis en ligne une partie des **sources** (lien 26) du jeu *Elite*, en langage C.

Pour avoir un aperçu de cette génération de noms aléatoires d'étoiles, il est possible d'expérimenter le Générateur Aléatoire d'Univers (lien 27) présenté dans un autre sujet (lien 28), enrichi de cette fonctionnalité, en passant la souris sur les divers lieux générés. En voici quelques exemples :

Bissole

Xesoleza

Sosolege

Bigecege

Soxelebi

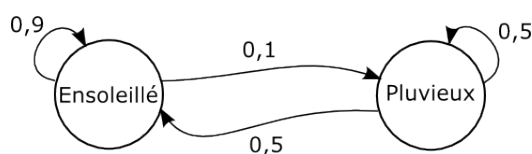
Dans un contexte spatial où les noms des étoiles peuvent être très exotiques, l'algorithme d'*Elite* peut suffire, même s'il y a toujours la possibilité de raffiner. Cependant, pour d'autres contextes, comme celui de la génération de prénoms ou de noms de villes, les noms issus de l'algorithme d'*Elite* ne feraient pas longtemps illusion.

Pour cela, d'autres méthodes sont nécessaires.

## 3 Chaînes de Markov

### 3.1 Théorie

Une chaîne de Markov est un processus stochastique mettant en jeu un ensemble d'états et des probabilités de transitions entre ces états dont le but est de modéliser les phénomènes où certaines séquences d'états ont davantage de chances de se produire que d'autres.



Exemple de chaîne de Markov

Sur le schéma ci-dessus, si le processus se trouve dans l'état *Ensoleillé*, alors il a 90 % de chances d'y rester (boucle) et 10 % de passer à l'état *Pluvieux*. Si le processus se trouve dans l'état *Pluvieux*, alors il a 50 % de chances d'y rester et 50 % de chances de passer à l'état *Ensoleillé*. Intuitivement, on devine que le processus aura tendance à enchaîner successivement davantage les états *Ensoleillé* que les états *Pluvieux*.

Une version animée de l'exemple précédent peut être visualisée sur le blog de Setosa : lien 29.

L'utilisation de chaînes de Markov comporte en général trois grandes phases. La première consiste à générer un modèle probabiliste à un phénomène à partir de l'expérience. Par exemple, à partir de



l'historique des relevés météorologiques, il est possible d'établir la probabilité de transition entre les différents temps (par exemple : ensoleillé, nuageux, pluvieux) à partir d'un temps donné en supposant que la météo du lendemain ne dépend que de la météo du jour concerné, sans influence des jours précédents.

La seconde phase consiste à appliquer ce modèle à partir d'une situation donnée pour prédire la probabilité des états futurs. Par exemple, à partir d'un modèle météorologique markovien, il est possible d'estimer la probabilité de beau temps et de mauvais temps en fonction de la météo du jour. Autre exemple, le *PageRank* de Google qui serait basé sur une chaîne de Markov, ne serait que la probabilité de transition vers une page donnée à partir d'une page quelconque.

Enfin, la troisième phase consiste à considérer le problème à l'envers qui est à partir d'une succession d'états, savoir si c'est une succession possible d'états dans une chaîne de Markov prédéfinie. Par exemple, à l'aide d'un modèle météorologique markovien, disons de la ville de Paris, il est possible d'analyser des relevés météorologiques sur une année et voir s'ils correspondent à une succession possible de temps à Paris. Selon les résultats, les relevés peuvent être déclarés compatibles avec la météo de Paris ou au contraire incompatibles ou improbables. En principe, un modèle météorologique markovien conçu pour Paris devrait déclarer comme incompatibles ou improbables les relevés météo de Tombouctou ou ceux d'Oslo et par contre accepter les relevés pris sur Paris. Notons que cette troisième phase s'inscrit plus généralement dans le domaine de la reconnaissance de formes où les chaînes de Markov sont notamment utilisées dans la reconnaissance vocale. Nous n'aborderons pas cette troisième phase dans cet article.

Aussi, une chaîne de Markov qui est à la base un objet mathématique fait appel à un certain formalisme, notamment celui de l'algèbre linéaire, que nous n'aborderons pas non plus ici. Pour avoir plus de développement sur les aspects théoriques des chaînes de Markov, le lecteur peut par exemple consulter le cours à ce sujet de Marc Lelage (INRIA/ENS) : [lien 30](#).

### 3.2 Mise en œuvre

Il est possible de faire appel aux chaînes de Markov pour la génération aléatoire de noms dans la mesure où l'on peut voir cette génération aléatoire comme une sorte de prédiction réalisée à partir d'un modèle probabiliste que nous aurons préalablement construit à partir d'une liste de noms.

La première phase, l'élaboration du modèle probabiliste, consiste à calculer et à stocker les probabilités de transition d'une lettre à une autre. Pour cela, notons le début d'un nom de la liste par le symbole

$\hat{\ }^$  qui sera le caractère de départ de n'importe quel nom et permettra d'initialiser la chaîne de Markov. De même, la fin d'un nom de la liste sera marquée par le symbole  $\$$ . Par exemple le prénom *jeanne*, sera interprété dans notre modèle comme *jeanne\$*.

Le but est de comptabiliser les occurrences de toutes les paires de lettres présentes dans la liste. Chaque paire représentant la transition de la première lettre de la paire vers la seconde. Par exemple la paire *ab* correspond à la transition entre la lettre *a* et la lettre *b*. Pour le cas d'une liste ne comportant que le nom *jeanne\$*, on aurait les paires suivantes avec leur occurrence associée :

```

 $\hat{\ }j$  : 1
je : 1
ea : 1
an : 1
nn : 1
ne : 1
e$ : 1

```

Nous pouvons déjà interpréter la chaîne de Markov sous-jacente par rapport à ces occurrences :

- si on se trouve à l'état  $\hat{\ }$  (c'est-à-dire en début de mot), la lettre suivante doit forcément être un *j* ;
- si on se trouve à l'état *j*, la lettre suivante doit forcément être un *e* ;
- si on se trouve à l'état *e*, il y a 50 % de chances pour que la lettre suivante soit un *a*, et 50 % de chances que ce soit un  $\$$  (c'est-à-dire la fin d'un mot) ;
- si on se trouve à l'état *a*, la lettre suivante doit forcément être un *n* ;
- si on se trouve à l'état *n*, il y a 50 % de chances pour que la lettre suivante soit un *n*, et 50 % de chances que ce soit un *e* ;
- si on se trouve à l'état  $\$$ , la chaîne de Markov est terminée.

Par exemple, cette chaîne de Markov pourrait produire les noms *je\$* ou encore *jeannnnneane\$*.

En JavaScript/TypeScript, il est possible de stocker les occurrences dans l'ersatz de tableau associatif qu'est un objet. Si on reprend les occurrences précédentes, cela pourrait donner ceci en TypeScript :

```

1 transitions = {
2   "~j": 1
3   "je": 1
4   "ea": 1
5   "an": 1
6   "nn": 1
7   "ne": 1

```



```
8 "e$": 1
9 }
```

Maintenant si on ajoute à la liste un nouveau nom, par exemple *alexandre\$*, voici ce que donnerait les nouvelles occurrences de paires :

```

j : 1
je : 1
ea : 1
an : 2
nn : 1
ne : 1
e$ : 2
â : 1
al : 1
le : 1
ex : 1
xa : 1
nd : 1
dr : 1
re : 1

```

La nouvelle chaîne de Markov sous-jacente à ces occurrences nous indique entre autres que si on se trouve à l'état *a*, il y a 66 % (2/3) de chances pour que la lettre suivante soit un *n*, et 33 % (1/3) de chances que ce soit un *l*. Compte tenu d'un nombre d'états et de transitions plus élevé que précédemment, les mots possibles que peut produire cette chaîne de Markov sont plus nombreux.

Comme il ne serait pas possible de calculer à la main l'ensemble des occurrences de paires dans une liste un minimum consistante, une routine peut évidemment faire le travail à notre place. En TypeScript, cela pourrait donner ceci :

```

1 function computeTransitions(database:
2   string[]): any {
3   var transitions = {};
4   for (var i = 0; i < database.length; i
5     ++ ) {
6     var word = database[i];
7     var prev = "^";
8     var next = "";
9     for (var j = 0; j < word.length; j++
10      ) {
11      next = word.substr(j, 1);
12      updateTransitions(transitions,
13        prev, next);
14      prev = next;
15    } // for j

```

```

14   next = "$";
15   updateTransitions(transition, s prev,
16     next);
17 } // for i
18
19 return transitions;
20 } // computeTransitions
21
22 function updateTransitions(transitions:
23   Object, prev: string, next: string)
24 {
25   var key = prev + next;
26   if (transitions.hasOwnProperty(key)) {
27     // transition déjà existante
28     transitions[key]++;
29   }
30   else { // new transition
31     transitions[key] = 1;
32   }
33 } // updateTransitions

```

On suppose que notre liste de noms, le paramètre *database*, est fournie sous la forme d'un tableau.

Pour l'exemple, nous pourrions considérer la liste de noms suivante :

```

1 var database = "agnès alain albert
2   alexandre annabelle anne"
3   "baptiste béatrice benoit
4   carl caroline cécile"
5   "christine christophe
6   emmanuel emmanuelle
7   emilie"
8   "éric eve frédéric
9   gaspard henri
10  henriette"
11  "isabelle jean jeanne
12  jennifer joseph léa
13  louis"
14  "marc marie marion maxime
15  michel nathalie
16  nicole"
17  "noémie olivia olivier
18  patrick paul philippe"
19  "
20  "pierre rené robert sé
21  bastien sophie sté
22  phane"
23  "stéphanie thierry".split
24  (" ");

```

Notez l'utilisation de la méthode *split* qui permet de convertir en tableau cette liste de noms séparés par des espaces.

Une chaîne de Markov étant donnée, l'algorithme pour générer aléatoirement une suite de lettres est assez intuitif et pourrait ressembler à ceci :

```

1 function randNameMarkov(transitions:
2   Object): string {
3   var state = "^";
4   var result = "";
5
6   while (state != "$") {
7     var newLetter = markov(state,
8       transitions);
9     result += newLetter;
10    state = newLetter;
11  } // while

```

```

11     return result.substr(0, result.length
12     - 1); // supprime $ en fin de ré
        sultat
12 } // randNameMarkov

```

où la fonction *markov* renvoie une nouvelle lettre choisie en fonction des probabilités de transition à partir de l'état passé en paramètre. Dans cette fonction *markov*, pour un état donné, nous pouvons nous baser sur la fonction de répartition de ses transitions pour générer une nouvelle lettre à partir d'un nombre aléatoire compris entre 0 et 1. La fonction de répartition (*cumulative distribution function* en anglais) peut schématiquement être comparée à une table de résolution de lancers de dés dans les jeux de rôle.

d66	Localisation des dégâts
11 à 14	Tête
15 à 24	Bras gauche
25 à 34	Bras droit
35 à 52	Poitrine
53 à 61	Jambe gauche
62 à 66	Jambe droite

Table de localisation des dégâts avec un d66 issue du jeu Heroes & Other Worlds

Voici une implémentation en TypeScript de la fonction *markov* :

```

1 function markov(state: string,
2   transitions: Object): string {
3   var cumDist = []; // Fonction de ré
4     partition
5   var sum = 0;
6
7   // Calcule la fonction de répartition
8     des transitions de l'état
9   for (var key in transitions) {
10    if (transitions.hasOwnProperty(key)
11      && // spécifique JavaScript
12        key.substr(0, state.length) ==
13          state) {
14      sum += transitions[key];
15      var nextState = key.substr(state.
16        length);
17      cumDist.push({ nextState:
18        nextState, cumFreq: sum });
19    }
20  } // for key
21
22  // Choisit au hasard le prochain état
23  var random = Math.floor(sum * Math.
24    random());
25
26  for (var i = 0; i < cumDist.length; i+
27    +) {
28    var nextState = cumDist[i].nextState
29      ;
30    var cumFreq = cumDist[i].cumFreq;
31    if (random < cumFreq) {
32      return nextState;
33    }
34  }

```

```

24 } // for i
25
26 return "";
27 } // markov

```

On notera qu'il serait préférable en termes de performance de stocker une bonne fois pour toutes la fonction de répartition des transitions pour chaque état, plutôt que de la reconstruire à chaque appel de la routine *markov*.

Le script exécutable de cette première approche est disponible [ici](#) : [lien 31](#).

En termes de résultats, voici ce qu'il est possible d'obtenir :

en  
ck  
chric  
riphendrivile  
lle  
mie  
an  
sébe  
st  
riolbéarle

On remarque qu'à moins de créer des règles ad hoc, il n'est pas simple de générer des noms d'une longueur donnée. Aussi, les prénoms générés ne sont pas pour le moment très vraisemblables.

Un axe d'amélioration peut consister à enrichir le corpus de texte servant à la construction de la chaîne de Markov, le contenu de la variable *database* si on se réfère au code mentionné précédemment. En effet, plus le corpus « d'apprentissage » est riche, et plus il y a de chances que les « bonnes » transitions, les plus fréquentes et donc les plus vraisemblables, soient choisies par l'algorithme.

Pendant, cela peut ne pas suffire et c'est pour cela qu'il existe une variante aux chaînes de Markov.

### 3.3 Extension

Il est possible d'améliorer la qualité des noms générés par une chaîne de Markov en rajoutant une notion de mémoire. Dans les exemples jusqu'à présent, les transitions ne dépendaient que de l'état courant et ignoraient les états antérieurs, ce qui en langue française peut poser quelques soucis. Par exemple, après la séquence *ch*, il est courant d'avoir une voyelle, tout comme c'est le cas de la séquence *qu*.

Cet aspect peut être pris en compte en considérant non plus des transitions entre une lettre vers

une autre, mais entre une paire de lettres vers une autre lettre. Par exemple pour l'exemple de la séquence *ch* suivie d'une voyelle on pourrait avoir les transitions suivantes :

ch -> a

ch -> e

ch -> i

ch -> o

ch -> u

ch -> y

Plus concrètement, cela revient à ne plus dénombrer les occurrences des paires de lettres dans une liste de noms, mais les occurrences de triplets de lettres (ou trigrammes). On parle alors de chaîne de Markov d'ordre 2, correspondant aux deux états nécessaires pour considérer une transition ; sachant que les chaînes de Markov simples sont des chaînes de Markov d'ordre 1 où un seul état, l'état courant, suffit pour connaître les probabilités de transition vers un nouvel état.

On remarque dès lors qu'il est possible de généraliser à un ordre quelconque afin d'améliorer la vraisemblance des noms générés, tout en ayant à l'esprit que le nombre de transitions possibles et donc à mémoriser (et à parcourir selon les implémentations), augmente exponentiellement à mesure que l'ordre est élevé. Aussi, autre aspect pernicieux, à mesure que l'ordre est élevé, la vraisemblance des noms générés s'accroît au point qu'à un moment les noms peuvent ne plus être que des noms déjà existants, sans nouveauté. Ce revers qualitatif est assez analogue au phénomène du surapprentissage dans le domaine de l'intelligence artificielle. Mais c'est un

## 4 Conclusion

Nous venons de voir une méthode très simple de génération aléatoire de noms, celle du jeu *Elite*, basée sur une suite de paires prédéfinies et équiprobables, et une autre méthode, celle des chaînes de Markov, légèrement plus sophistiquée, et qui fournit des résultats plus généraux, applicables à divers contextes y compris celui de la génération aléatoire de phrases, à condition de disposer d'un corpus initial d'exemples de taille et de qualité suffisantes.

Il existe de nombreuses pistes pour améliorer les résultats des chaînes de Markov présentées ici.

Retrouvez l'article de *yahiko* en ligne : [lien 35](#)

autre sujet.

Sur la base des prénoms français, une chaîne de Markov d'ordre 4 ne génère pratiquement pas de noms originaux, alors qu'une chaîne de Markov d'ordre 3 donne des résultats plus intéressants.

On peut trouver une démonstration en ligne des chaînes de Markov sur les corpus des villes françaises (ordre 4) et des villes britanniques (ordre 4), ainsi que sur les prénoms français (ordre 3) : [lien 32](#).

Voici un résultat possible issu de la démonstration en ligne avec les prénoms français :

Stéphaël

Cécile

Émille

Armelle

Léonaticienne

Ariste

Marthée

Henri

Corin

Emence

On constate en premier lieu que la vraisemblance ici est meilleure qu'avec une chaîne de Markov simple sur un corpus d'apprentissage réduit. On note aussi que l'algorithme reprend des prénoms déjà existants (e.g. Cécile), mais fait preuve tout de même d'un peu d'imagination (e.g. Stéphaël, Léonaticienne) et c'est ce que nous souhaitions.

À noter que le code source de la démonstration en ligne est disponible sur mon compte GitHub : [lien 33](#).

Par exemple, nous pourrions contraindre l'algorithme à produire des noms d'une longueur déterminée. En plus des transitions issues du corpus initial d'exemples, nous pourrions aussi imposer des contraintes sur la structure même des séquences possibles via l'introduction de grammaires formelles.

Enfin, les chaînes de Markov présentées succinctement dans cet article pour la génération aléatoire de noms peuvent également être utilisées pour d'autres aspects d'un jeu vidéo, comme en intelligence artificielle : [lien 34](#).

# 2D/3D/Jeux

## Les dernières news 2D/3D/Jeux

# Modélisation et optimisation des décisions dans la conception de jeux - Introduction

Cet article est le premier d'une série sur l'application de techniques de modélisation et d'optimisation des décisions dans la conception de jeux. Il présente le parcours de la conception d'un jeu et précise les endroits où ces techniques de modélisation et d'optimisation se révèlent utiles.

## 1 De la recherche, pas d'itérations !

La plus grande partie de la conception d'un jeu (*game design*) est un processus de recherche. Le concepteur évalue un grand nombre de configurations possibles afin de résoudre un problème de conception — déterminer la manière de connecter les pièces d'un donjon, les caractéristiques et capacités des différents types d'agents du jeu, les « nombres magiques » qui gouvernent l'efficacité des unités dans un système de combat, voire la combinaison de caractéristiques que le jeu aura dès le début.

Tout comme un personnage dirigé par une intelligence artificielle utilisera un système de recherche de chemin pour naviguer dans le monde du jeu, la conception implique une certaine navigation dans un espace abstrait constitué de toutes les configurations possibles en prenant une configuration de base et en la modifiant de manière itérative : le concepteur regarde attentivement une partie de la conception — le système de combat, une partie du monde, un arbre des technologies dans un jeu de stratégie, etc. — et tente de trouver une manière de l'améliorer en changeant cette configuration.

Les concepteurs aiment utiliser le terme « itération » pour décrire ce procédé, mais « recherche » le décrirait mieux. La vérité est que, lors de l'« itération », le concepteur procède à des expériences avec le jeu en cours de développement. En utilisant son expérience, il effectue de petites modifications qui transformeront la configuration actuelle en une nouvelle qui — tout du moins, il l'espère — remplira mieux les critères.

Ces « itérations » ne ressemblent généralement pas aux changements linéaires qui apparaissent dans les « itérations » propres à la programmation informatique ; elles ressemblent bien plus à une recherche

à travers un labyrinthe, avec beaucoup de tournants brusques et, parfois, des retours en arrière. Ces déplacements mènent plus près de l'objectif, mais sans donner l'impression franche d'avoir amélioré le jeu. Parfois, une décision qui devait améliorer le jeu a des défauts qui n'étaient pas prévus, il est alors nécessaire de revenir en arrière.

La conception de jeux est une discipline extrêmement difficile : la situation est similaire à une pièce sombre, remplie d'objets pointus, à travers laquelle il est extrêmement difficile de naviguer sans danger à l'écart des sentiers battus. Le voyageur subit presque toujours les mêmes douloureuses blessures, en particulier s'il se déplace trop vite. Malheureusement, il dispose de peu d'outils pour éclairer cette pièce sombre, de très peu de techniques bien définies et disciplinées pour effectuer cette recherche.



Cette pièce noire est la raison pour laquelle il est nécessaire d'« itérer » : il n'est pas toujours possible de connaître les ramifications d'une décision tant qu'elle n'est pas tentée. En d'autres mots, le concepteur doit chercher (même Will Wright en parle, en faisant référence à une « recherche dans un espace de solutions », au cours de sa conférence à la GDC 2004 : [lien 36](#)).

Par conséquent, la phase de conception limite fortement la productivité de l'équipe de développement et est une source majeure d'imperfections : elle concentre les risques pour le développement d'un jeu. Un très grand nombre d'équipes se trouve limité par de mauvaises décisions de conception, des déferlements de créativité non concertés, de lents glissements dans les fonctionnalités, une mauvaise perception du marché cible ou d'autres problèmes concernant la conception qui ont mené à des problèmes de qualité du produit.

Avec tous les dangers qui rôdent lors de l'expérimentation de la conception du jeu, il n'est pas

étonnant que de nombreux éditeurs et développeurs évitent les risques, qu'elle qu'en soit la forme, en préférant explorer les zones proches de genres, de licences, de conventions bien établies et explorées en long et en large, plutôt que de courir les risques si bien connus de l'innovation avec des retours relativement inconnus. Explorer la pièce sombre leur semble simplement trop risqué.

Comment changer cette attitude ? Au lieu de simplement éviter l'innovation, il serait mieux de trouver des méthodes pour améliorer les compétences de conception, de construire des outils pour que l'innovation soit plus sûre et plus efficace.

## 2 À propos de cette série

Cet article est le premier d'une série qui présentera la modélisation des décisions, un ensemble d'outils pour décomposer les décisions en des modèles formels qui peuvent être explorés pour trouver la meilleure solution.

La modélisation et l'optimisation des décisions sont des techniques fréquentes dans le management, la finance, la planification avancée de projets et bien d'autres domaines encore, afin d'améliorer le processus décisionnel. Ces techniques impliquent de résoudre les difficiles problèmes de décision et d'optimisation en recherchant les solutions possibles bien plus vite que ne le pourraient des humains.

Malgré tous les bénéfices potentiels, la modélisation et l'optimisation des décisions semblent relativement inconnues parmi les concepteurs de jeu de l'industrie. Un sondage récent sur un forum populaire a montré qu'à peine vingt-cinq pour cent des participants avaient entendu parler de la modélisation des décisions : à peine huit pour cent l'avaient déjà utilisée. Une étude similaire sur Facebook, envoyée directement aux concepteurs, a donné des résultats presque identiques avec un nombre proche de répondants.

## 3 Définition

Qu'est-ce que la modélisation des décisions ? En des termes simples, **la modélisation des décisions est le processus de simulation d'une décision et de recherche automatique de sa solution.**

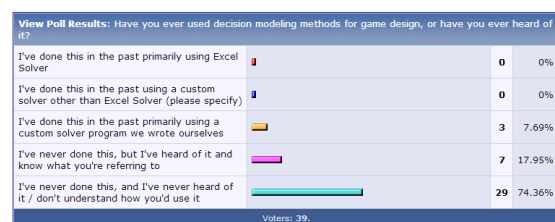
Ainsi, la première étape est de définir une sorte de décision de conception du jeu, de tenter de prendre en compte tous les facteurs d'intérêt qui l'influencent, puis de construire un modèle qui représente de manière précise la décision, en définissant des variables d'entrée (de *décision*) et une seule variable de sortie. Ensuite, un algorithme se charge d'effectuer une recherche pour les valeurs optimales des variables de décision qui produisent la meilleure

Utilisée correctement, la modélisation des décisions peut améliorer de manière significative un certain nombre d'aspects du processus de conception :

- elle peut aider à optimiser la configuration de systèmes spécifiques ou à trouver les valeurs optimales des paramètres du jeu ;
- elle peut apporter un éclairage sur des décisions comme les combinaisons de fonctionnalités à inclure dans le jeu ;
- elle peut aider à modéliser les décisions qu'un joueur peut prendre, particulièrement en termes d'identification des stratégies dominantes.

Cette série fournira des exemples dans chacune de ces trois catégories.

sortie. Si tout se passe bien, ce procédé permet d'explorer bien plus de solutions possibles qu'à la main ou par l'imagination. Il ne peut cependant pas s'appliquer à tous les cas : cette technique est appropriée pour une série de problèmes, mais pas tous. Pour ceux-là, le résultat est souvent bien meilleur, arrive plus rapidement et, dans certains cas, donne une solution à des problèmes qui, autrement, resteraient insolubles.





Tout au long de la série, des ensembles de *contraintes* seront définis : elles agiront en barrières qui s'assurent que le modèle reste valide. Ces

contraintes peuvent limiter la plage de valeurs ou le temps des variables d'entrée ou n'importe quel autre aspect du modèle.

## 4 Pourquoi construire des modèles ?

Les joueurs de Sid Meier's Civilization (lien 37) se sont probablement fait la réflexion suivante :

« Un instant... Quelle est la bonne manière de démarrer une ville ? D'abord construire un monument, puis un grenier ? Ou bien le grenier d'abord ? Ou peut-être le temple en premier, puis le grenier ? Quelle est la meilleure décision ? Existe-t-il seulement une manière de répondre à cette question ? »

Des pensées similaires peuvent venir à l'idée d'un joueur de jeu de stratégie en temps réel. Équilibrer les paramètres des différentes unités dans ce type de jeu est particulièrement difficile. Il serait très intéressant d'avoir un système pour accélérer la résolution de ce problème d'équilibre, capable de répondre à des questions sur le système de combat sans devoir jouer une seule partie. Et s'il était possible de poser à ce système des questions comme « Combien de spadassins faut-il pour infliger une défaite à deux piquiers et trois archers ? » ou encore « Quelle est la combinaison la moins chère d'archers et de catapultes pour mettre à terre une tour de garde ennemie ? ».

En fait, c'est possible !

En modélisant ces problèmes de la bonne manière, des outils automatiques d'optimisation pourront chercher la meilleure réponse qui satisfasse à ces critères, sans devoir jouer des milliers de parties.

Voici un exemple d'un problème du même genre, un exemple qui sera réutilisé dans les futurs épisodes de cette série : un jeu nommé SuperTank. Le joueur contrôle un gigantesque tank rempli de technologies de pointe, avec lesquelles il doit défaire d'autres supertanks. Avant chaque bataille, il peut sélectionner la combinaison exacte d'armes pour équiper son tank.



Le joueur possède cent crédits qu'il peut dépenser sur son armement. Son tank peut transporter au plus cinquante tonnes d'armes et dispose de trois « emplacements critiques » où il peut installer des armes spéciales à haute puissance.

Le jeu propose cinq types d'armes, le joueur peut en sélectionner autant qu'il veut de chaque type et même passer certains équipements.

	Dommages	Poids	Coût	
Mitrailleuses	2	1	5	0
Roquettes	8	3	12	0
Mégaroquettes	15	10	16	1
Lasers	7	4	9	0
Ultralasers	20	16	18	1

Petit problème : comment équiper le tank pour causer le plus de dommages possible (en supposant qu'il s'agit de dommages causés à l'ennemi par seconde, peu importe la vitesse de tir) ? Par hypothèse, toutes les armes auront la même portée, le même arc de tir, la même précision, de telle sorte qu'elles sont toutes identiques à tout point de vue, à l'exception des données du tableau ci-dessus.

Vite ! Combien de mitrailleuses, de roquettes, de lasers devraient équiper le tank ? Quelle combinaison d'une ou plusieurs armes inflige le plus grand dommage sans dépasser le budget, le poids total et le nombre d'emplacements critiques disponibles ?

Essayez de résoudre ce problème à la main, voire avec une calculatrice.

Alors ?

Si vous avez essayé, vous avez pu remarquer que ce problème est particulièrement difficile, bien plus qu'attendu. Il est probablement possible de s'en sortir à coups d'équations alambiquées, mais ce n'est pas le domaine de prédilection des concepteurs de jeux vidéo.

D'ailleurs, comment évoluerait la solution pour d'autres paramètres ? Que se passerait-il si le tank pouvait porter dix tonnes supplémentaires (soit un total de soixante tonnes) ? Si le joueur avait cent dix ou nonante crédits à dépenser ? S'il avait quatre emplacements critiques ou seulement deux ?

Imaginez un instant un système qui pourrait calculer instantanément la combinaison d'armes qui cause les plus grands dommages pour n'importe quelles valeurs de ces trois paramètres (poids total, crédits disponibles, nombre d'emplacements critiques) : il suffirait d'introduire les données des armes, les trois caractéristiques du tank pour obtenir, directement, la meilleure manière de l'équiper. Ne serait-ce pas magique ?

Un tel système pourrait donner des réponses immédiatement à une série de questions fort utiles pour



la conception du jeu.

- Comment évolue la charge optimale en modifiant les paramètres du tank? Ceux des armes?
- Quels dommages peut, au plus, infliger un tank pour des paramètres donnés?
- Les quatre paramètres de charge arme sont-ils appropriés et équilibrés?
- Y a-t-il des armes trop puissantes et utilisées trop fréquemment? Si une arme est tellement puissante que la meilleure décision l'utilise toujours, alors l'embarquer n'est pas réellement un choix. Dans ce cas, il faudrait probablement retirer l'arme du jeu ou la rééquilibrer de telle sorte qu'il existe des cas où elle n'est pas utile.
- Y a-t-il des armes sous-utilisées, rarement ou jamais utilisées? Comme précédemment, si une arme est tellement inutile que, peu importe le cas, la meilleure décision est de ne

pas l'utiliser, cette décision n'a pas vraiment de sens non plus. Dans ce cas, il faudrait soit retirer l'arme du jeu, soit la rééquilibrer pour qu'il existe des situations où elle est utile.

Il s'agit de décisions très importantes sur la conception du jeu, toute personne impliquée devrait vouloir la réponse à ces interrogations : ces résultats sont extrêmement intéressants pour équilibrer SuperTank.

Quelques paragraphes ont suffi pour décrire un problème remarquablement difficile à résoudre manuellement, mais les outils de base d'Excel peuvent le résoudre très facilement. D'ailleurs, la construction d'un modèle de décision pour ce problème sera l'objet d'un prochain article où les questions précédentes trouveront leur réponse : cette construction prendra à peine quelques minutes, malgré la difficulté du problème ; avec un peu de travail, il devient possible de créer un outil puissant pour explorer l'espace de conception.

## 5 Feuille de route

Cette série illustrera quelques autres exemples plus compliqués et fournira également quelques feuilles de calcul de référence, afin de répéter les manipulations sans utiliser autre chose qu'Excel. Parmi ces exemples, notamment :

- un exemple de combat, relativement simple, pour un jeu de stratégie ;
- un modèle pour optimiser les coordonnées de plusieurs téléporteurs dans un jeu spatial massivement multijoueur, en prenant en compte tant les positions de chaque trou que de la répartition de la population ;
- un modèle pour déterminer le niveau de taxe dans une ville simplifiée, afin d'équilibrer le niveau de bonheur de la population par rapport aux taxes et aux revenus, dans un jeu de type 4X comme Sid Meier's Civilization ;
- un modèle pour choisir la manière d'assigner des sorts et d'autres caractéristiques à des classes de personnages dans des jeux massivement multijoueurs ;
- un modèle d'optimisation pour déterminer

l'ordre optimal des constructions sur une planète colonisée dans un jeu de stratégie de type 4X, comme le classique Master of Orion ;

- un exemple de choix de fonctionnalités pour un jeu en utilisant un modèle de décision pour effectuer le meilleur compromis.

En général, les exemples commenceront par chercher des stratégies optimales pour un joueur dans un sous-système spécifique du jeu, avant de progresser vers des modèles de décision qui aident à optimiser les paramètres pour tous les systèmes d'un jeu et les combinaisons de fonctionnalités.

Dans chaque cas, le problème sera décrit et modélisé dans Excel, avant une résolution avec le solveur inclus. La feuille de calcul sera aussi disponible pour chaque exemple afin de faciliter les tests et l'expérimentation sur chacun des modèles.

Cependant, la représentation sous-jacente, que ce soit une feuille de calcul ou du code dans un langage de programmation, n'a aucune espèce d'importance : le point important n'est pas de résoudre le problème avec le solveur d'Excel ou en C, C++, Java ou autre, mais bien la manière de modéliser la situation.

## 6 Pourquoi utiliser des modèles de décision ?

Certains lecteurs pourraient, à présent, se montrer incrédules : construire des modèles de décision semble requérir beaucoup de travail, pourquoi le faire quand des joueurs seront prêts à tester les différentes décisions sur des préversions du jeu ?

Tout d'abord, il est important de remarquer que

**la modélisation des décisions ne s'applique pas à tous les problèmes.** Certains sont trop compliqués, trop difficiles à modéliser avec ces techniques ; d'autres aspects de la conception, comme l'esthétique ou l'amusement, sont difficiles voire impossibles à modéliser avec des nombres. La modélisa-

tion des décisions n'élimine pas d'emblée les phases de test externes ou internes.

Cela dit — et ce devrait être clair à la fin de la série —, la méthode de modélisation et d'optimisation des décisions donne aussi un ensemble d'outils unique et très puissant, qui peut, en tout ou en partie, résoudre bien des problèmes qui ne pourraient pas trouver de solution autrement. Ces outils peuvent fournir des réponses, une compréhension plus fine de toute sorte de questions de conception, ce qui ne serait pas possible autrement. Comme pour tout outil, le praticien fait son choix prend l'outil le plus approprié.

Dans de nombreux cas, les modèles de décision ne sont pas appropriés, trop encombrants pour être utiles. Cette série montrera néanmoins qu'ils se révèlent extrêmement pratiques dans d'autres cas : si les décisions au début du développement du jeu sont bien prises, en éliminant des problèmes avant même d'atteindre les phases de test, il est plus probable que le jeu complet sera solide, amusant et sans défaut.

Le programmeur fait face au même dilemme : son métier est difficile, mais il dispose d'un grand nombre d'outils pour trouver des défauts avant même que le code arrive aux tests. Notamment, ils ont des compilateurs qui crient en permanence dès la première faute de frappe ; les pratiques de programmation défensive montrent les défauts logiciels ; les revues de code identifient les défauts du code d'autrui et peuvent servir à éliminer de mauvaises habitudes de programmation ; les outils de profilage et d'analyse statique servent également à trouver les problèmes de performances et d'autres ennuis.

Les concepteurs n'ont pas ce genre d'outils. Leur travail est probablement aussi difficile, mais ils ne disposent pas de compilateur pour éviter les erreurs

## 7 Pas de feuille de calcul : des modèles !

Cette série d'articles s'oriente principalement vers les concepteurs — tous les concepteurs, qu'ils aient une expérience plus artistique, de programmation, narrative ou de jeu de plateau. Ils resteront donc relativement simples et respecteront ces quelques promesses :

- *pas de code*. Aucun article ne contiendra du code : tous les exemples utiliseront Excel et son solveur. Cependant, il est important de remarquer que cette série ne concerne pas l'utilisation de feuilles de calcul ou d'Excel, mais bien la modélisation et l'optimisation des décisions. Tout ce qui est présenté peut très bien être réalisé avec d'autres outils dans un langage de programmation, parfois même plus facilement ;
- *pas de mathématiques* — tout du moins, rien

de syntaxe ; ils n'ont pas de profileur, d'outils de débogage ou d'analyse statique. Il est impossible de faire des revues de code, puisqu'il n'y a pas de « code » à proprement parler. Ils écrivent des spécifications, de la documentation sur les choix de conception... et c'est à peu près tout. Ils peuvent partager ces documents avec l'équipe et espérer avoir des retours de qualité, mais, en grande partie, il est nécessaire d'entrer dans le jeu pour voir si ces décisions tiennent la route.

Par conséquent, la conception de jeux est incroyablement risquée, elle prend beaucoup de temps et d'argent. Tout comme la programmation, cependant, l'erreur humaine est naturelle et fait partie intégrante du processus : il est *nécessaire* de disposer d'autant d'outils de bonne facture que possible pour protéger tant les concepteurs que leurs projets.

La route est longue avant d'avoir des outils de conception qui aideront à l'exploration de l'espace de conception du même niveau que les compilateurs, débogueurs, profileurs, outils d'analyse statique, qui aident les programmeurs à explorer l'espace de développement. Cependant, de nouveaux outils pour la conception et la résolution dans quelques jeux apparaissent, notamment un vérificateur de jouabilité pour une variante de Cut the Rope nommée Cut the Rope : Play Forever ([lien 38](#)). Il faut aussi compter sur le système de conception de jeu abstrait Ludi ([lien 39](#)), qui a notamment généré le jeu de plateau Yavalath, mais encore l'assistant d'équilibrage automatique pour City Conquest : [lien 40](#).

La modélisation des décisions peut aider à franchir quelques étapes pour accroître et prolonger l'intelligence des concepteurs avec quelques outils automatiques. De toute façon, avec le choix d'avoir ces outils ou pas, pourquoi choisir de ne pas les avoir ?

de compliqué. Cette série ne comportera que peu de mathématiques et, en tout cas, pas plus que les opérations arithmétiques de base : addition, soustraction, multiplication, division, peut-être à l'occasion une racine carrée. Les lettres grecques seront strictement interdites ;

- *pas de feuille de calcul à quatre dimensions*, deux suffiront largement : [lien 41](#).

Pour les concepteurs, cette série devrait donner tous les outils nécessaires pour créer des modèles de décision, sans devoir tenter d'écrire du code ou compter sur des programmeurs. Pour les programmeurs, elle devrait donner un guide relativement rapide pour la programmation de modèles de décision dans n'importe quel langage de programmation.

Ces articles ne sont prévus que comme des points de départ, des concepts à utiliser pour construire

des modèles dans Excel, n'importe quel autre outil d'optimisation ou dans un solveur à écrire soi-même. Les feuilles de calcul sont un bon départ, mais ces

modèles de décision seront plutôt prévus comme des tremplins pour des modèles plus sophistiqués et plus riches à intégrer dans l'architecture du jeu.

## 8 Avertissements

Avant d'entrer dans le vif du sujet, quelques avertissements semblent nécessaires. La modélisation et l'optimisation des décisions ne fournissent d'aucune manière un système complet pour la conception d'un jeu et ne le prétendent. Cette méthode se présente comme un outil, utile pour aborder certains aspects du processus de conception et, comme tout outil, a ses limites.

Voici les limites dont il importe d'être conscient :

- **ces outils sont faciles à mal utiliser.** Comme tout outil, les modèles de décisions peuvent être utilisés de manière inappropriée ou incorrecte ; un modèle de décision incomplet ou partiellement faux peut mener à des conclusions incorrectes. Tout comme un logiciel, plus le modèle est grand, plus il est probable qu'il soit incorrect. Il est aussi très facile de mal interpréter les résultats d'un modèle ou de construire un modèle incomplet qui ne prend pas en compte avec une précision suffisante le contexte des décisions à prendre ;
- **ces outils sont parfois compliqués.** Certains problèmes de conception sont trop complexes pour être modélisés de la sorte. De nombreux problèmes ont beaucoup de parties mouvantes ou sont trop intégrés avec d'autres aspects du jeu pour être représentés de manière utile dans une feuille de calcul Excel indépendante. Dans ces cas, il faut décider de modéliser seulement une partie du système (ce qui peut donner un modèle invalide ou imprécis), de construire un modèle complet intégré dans le jeu lui-même (ce qui représente une grande quantité de travail) ou bien s'abstenir d'utiliser ces outils ;
- **tout ne peut pas être modélisé.** Les modèles de décision ne peuvent pas dire si une décision sera amusante pour les joueurs, si elle est esthétique, si elle s'intègre bien dans le contexte du jeu ou si elle présente au joueur une interface utilisable et accessible. Il n'existe bien souvent pas de manière de représenter ces préoccupations subjectives ou esthétiques dans un modèle. Cela signifie qu'il existe des limites aux cas d'utilisation des modèles de décision : ils sont bien plus utiles pour la conception des systèmes et l'optimisation des mécaniques de jeu et de ses dynamiques plutôt que de l'esthétique ;
- **ces outils ont leurs limites.** Toutes les solutions d'optimisation ont leurs limites, notam-

ment le solveur d'Excel : il est possible de créer des modèles qui possèdent des solutions valides, mais sont tellement complexes qu'aucun outil d'optimisation ne peut les trouver. Pour un nombre de variables suffisamment grand, le problème peut se révéler en dehors des capacités d'évaluation de toutes les combinaisons d'Excel : il doit alors utiliser d'autres techniques d'optimisation. Cette série montrera notamment des manières de simplifier l'expression des modèles pour que le solveur les appréhende plus facilement ; d'ailleurs, le développeur du solveur d'Excel, Frontline, fournit un solveur bien plus puissant pour des problèmes plus grands, mais il reste possible de créer des modèles que ce solveur ne peut pas résoudre ;

- **ces outils ne garantissent pas l'optimalité.** À cause des limites inhérentes à la technologie, pour des modèles complexes, il est impossible de garantir que la solution donnée est optimale. En passant plus de temps à optimiser, la solution sera de meilleure qualité ; en relançant le modèle depuis zéro, il est aussi possible de dire que la solution est proche de l'optimalité à un niveau de confiance raisonnable.

Finalement, le point le plus important : **l'utilisateur doit s'assurer de modéliser les bons éléments.** Tous les problèmes ne sont pas importants au point de nécessiter ce genre d'efforts, il faut s'attribuer des priorités et éviter de trop se concentrer sur des défauts mineurs du modèle en ignorant d'autres, bien plus importants.

De manière générale, certaines choses doivent être vraies pour que la modélisation des décisions soit utile. La décision en question doit être encapsulable dans un modèle discret et faire correspondre la décision à une seule valeur. En d'autres mots, l'entrée doit être un ensemble fini de décisions possibles, le modèle fournit une seule sortie, de telle sorte que maximiser ou minimiser cette sortie donne la meilleure solution.

Dans ces cas où des intérêts subjectifs ne peuvent pas être insérés dans le modèle, comme des considérations d'ordre esthétique ou de jouabilité, il faudra soit les séparer clairement du modèle de décision, soit utiliser ces techniques comme une première phase, soit complètement abandonner l'approche.

Pour modéliser des décisions dans une feuille de calcul, il y a aussi une limite sur la complexité des modèles. Si le jeu effectue des actions très complexes,

il n'est pas sûr qu'elles puissent être répliquées dans Excel. Il est important de garder à l'esprit, cependant, que ce n'est qu'une limite sur les modèles présentables à Excel, pas des modèles de décision eux-mêmes. Des solveurs spécifiques peuvent être implémentés dans le moteur de jeu, bien plus puissants que des feuilles de calcul : cette série a été écrite dans l'espoir d'inspirer des concepteurs à le faire.

D'un autre côté, toutes ces limitations ne sont pas suffisantes pour rendre la modélisation des décisions inutile. Même dans le cas où un problème est trop complexe, sa modélisation imparfaite peut donner un certain nombre de parties du système plus proches d'une configuration correcte, ce qui aide à trouver et à corriger un certain nombre de défauts dès le début du développement.

Même dans le cas où la modélisation ne permet pas de trouver la solution optimale à un problème, à cause de sa complexité ou de considérations hu-

maines subjectives comme l'esthétique, l'espace de recherche est fortement réduit, la solution imparfaite élimine des zones de moindre intérêt, ce qui diminue la complexité du problème à résoudre.

Finalement, même pour ceux qui décideront de ne pas utiliser la modélisation des décisions, qui ne tenteront jamais d'optimiser sur une feuille de calcul ou de construire leurs propres solveurs, une compréhension de ces techniques peut toujours aider en changeant la manière de penser les décisions de conception d'un jeu.

Cette série est une exploration : elle observera bien des exemples de problèmes de conception de jeux et explorera des manières de les modéliser et de les optimiser qui offrent de puissants outils de conception. L'espoir est que même les plus sceptiques, ceux qui se sentent moins à l'aise avec l'optimisation, explorent ces contrées par cette série, pour voir à quoi elle mène.

## 9 Conclusion

L'objectif final est de concevoir des jeux correctement.

La plupart des choix de conception sont subjectifs, sans « bonne » ou « mauvaise » réponse. Dans certains cas, cependant - bien plus nombreux que ce que l'on peut penser -, certains choix sont purement objectifs. Dans ces cas, le concepteur devrait vouloir connaître cette bonne réponse, au moins comprendre comment la définir et chercher cette solution, si elle existe.

La modélisation et l'optimisation des décisions sont des outils puissants qui vont dans cette direction, dans bien des cas. Ils devraient faire partie de la boîte à outils de chaque concepteur de jeux. Avec un peu de discipline, il devrait devenir clair que ces outils ont un potentiel peu exploité pour explorer la pièce noire de la conception de jeux plus vite et avec une sécurité accrue. Le reste de cette série en montrera différentes applications.

*Retrouvez l'article de **Paul Tozour** traduit par **Thibaut Cuvelier** en ligne : [lien 42](#)*



# Libres & Open Source



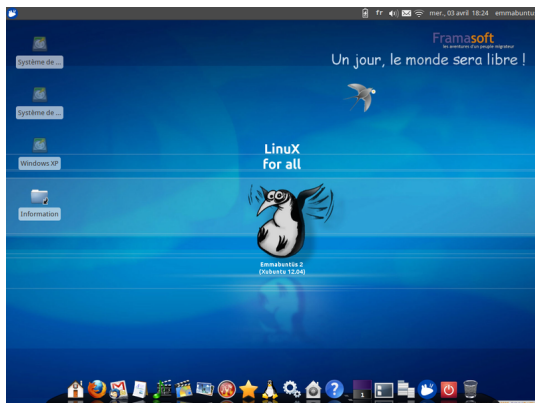
## Les dernières news Libres & Open Source

### Emmabuntüs : Ubuntu pour Emmaüs

Emmabuntüs ([lien 43](#)) est une distribution GNU/Linux ([lien 44](#)) construite avec Ubuntu 10.04 LTS puis Xubuntu 12.04 LTS pour reconditionner les ordinateurs donnés aux communautés Emmaüs et aux associations humanitaires. Une troisième version basée sur Xubuntu 14.04 LTS 32 bits et 64 bits est publiée depuis le 2 septembre 2014 : [lien 45](#).

Sans ce travail bénévole, ces machines seraient vouées au recyclage des matières premières.

**Emmabuntüs** est un mot-valise composé de Emmaüs et Ubuntu et se prononce Emma-boun-tous, comme Xubuntu se prononce Zou-boun-tou.



Elle contient un grand nombre de logiciels pour couvrir la plupart des usages d'une famille. Comme ASRI éducation, elle inclut la suite bureautique pour enfants de 6 à 12 ans OOo4kids. Elle est distribuée dans une image ISO de plus de 3 Go et tient sur un DVD ou une clé USB de 4 Go afin de pouvoir l'envoyer par La Poste en Afrique, dans les lieux dépourvus de connexion Internet.

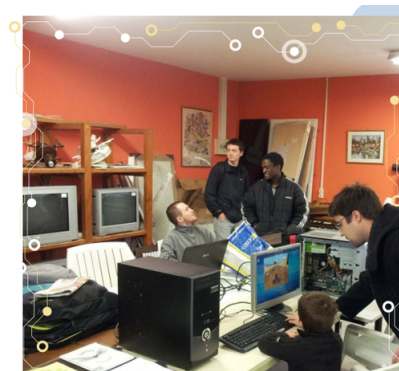


Elle a été adoptée par les Jerry-Clan de ce continent qui l'utilisent pour équiper leurs Jerrys, ces ordinateurs assemblés dans un bidon en plastique de vingt litres. Ce récipient est d'un usage très courant dans ces pays.

**Des magazines de la presse spécialisée ont publié des articles sur cette distribution GNU/Linux humanitaire.**

1. LINUX ESSENTIEL N° 33 : Emmabuntüs le reconditionnement pour tous ([lien 46](#)).
2. LINUX PRATIQUE HORS SERIE N°28 : Le reconditionnement pour les plus démunis ([lien 47](#)).
3. L'âge de faire mars 2014 : L'informatique au service de l'humanitaire ([lien 48](#)).
4. TERRA ECO septembre 2014 : Emmabuntüs ravive la mémoire des ordis ([lien 49](#)).

**Le groupe d'utilisateurs de logiciels libres le plus actif pour la présenter est sans doute Montpel'libre : [lien 50](#).**



Je constate que la plupart des associations qui reconditionnent des ordinateurs pour les donner ([lien 51](#)) ou les vendre à des personnes qui veulent s'initier à l'informatique utilisent la même base Ubuntu, Xubuntu. Ordi-solidaires utilise Debian, mais ne distribue pas d'image disque contenant ses suites logicielles : [lien 52](#). Handylinux ([lien 53](#)) est la seule distribution dérivée officielle de Debian à ma connaissance avec des atouts semblables à Emmabuntüs. Elle est même classée dans la catégorie vieux ordinateurs par Distrowatch ([lien 54](#)) alors qu'Emmabuntüs ne l'est pas.



Emmabuntüs est fournie avec l'environnement **Xfce**. Une icône dans le dock "All" permet d'installer **LXDE** pour avoir un environnement de bureau plus léger et une consommation de mémoire moindre, cependant Emmabuntüs ne pourra pas fonctionner sur un ordinateur doté d'un processeur Intel Pentium III à 800 MHz et 256 Mo de ram comme Lubuntu 14.04 LTS installé par Mikedafunk : [lien 55](#). Emmabuntüs fonctionne agréablement sur un ordinateur équipé d'un processeur **AMD Athlon XP 2000+**, **1 Go de ram**, **20 Go d'espace disque pour le système**, une carte graphique

*Commentez la news d'[ideefix](#) en ligne : [lien 58](#)*

**AGP 8x** pour exécuter les jeux comme SuperTux-Kart.



Pour débiter sous Linux, vous pouvez choisir le **pack simplifié d'applications** et le **HandyMenu** de HandyLinux ([lien 56](#)) ou bien télécharger Emmabuntüs ([lien 57](#)) avec son **dock** pour accéder à sa **panoplie complète de logiciels**.



## Les derniers tutoriels et articles

# Apprendre à utiliser GanttProject

Au cours de ce tutoriel, je ne vais pas vous apprendre à gérer un projet, mais je vais vous montrer comment utiliser GanttProject qui est un des outils libres de gestion de projet.

## 1 Définition

GanttProject est un logiciel libre, permettant de modéliser sous forme de Gantt la planification des différentes tâches qui sont nécessaires à un projet. Chaque tâche pourra être définie par sa durée et par les ressources qui pourront lui être affectées. Vous pourrez aussi obtenir des documents comme le PERT et le chemin critique.

Les logiciels de gestion de projet sont généralement utilisés pour automatiser la sauvegarde des tâches et la gestion du temps. La visualisation d'un Gantt permet par un simple schéma graphique de

savoir qui fait quoi en donnant l'état d'avancement du projet, de chaque tâche et sous-tâche, d'avoir la planification de l'ensemble et la charge de travail des équipes.

Les graphiques ainsi obtenus pourront être imprimés ou enregistrés dans différents formats (HTML, PDF ou images), vous aurez aussi la possibilité d'exporter les données vers un tableur, vous permettant ainsi de donner les différentes tâches aux personnes concernées.

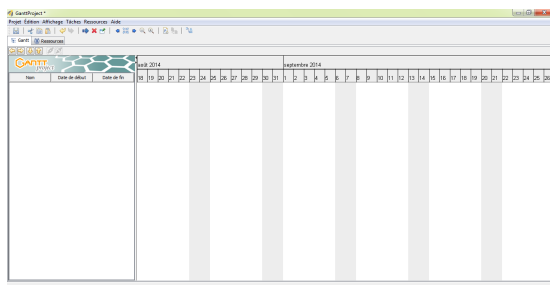
## 2 Installation

C'est un logiciel multiplateforme (Linux, Mac OS et Windows). La page de téléchargement se trouve sur le site de GanttProject : [lien 59](#). Si vous êtes sous Windows ou Linux, vous pourrez trouver une version portable de l'outil.

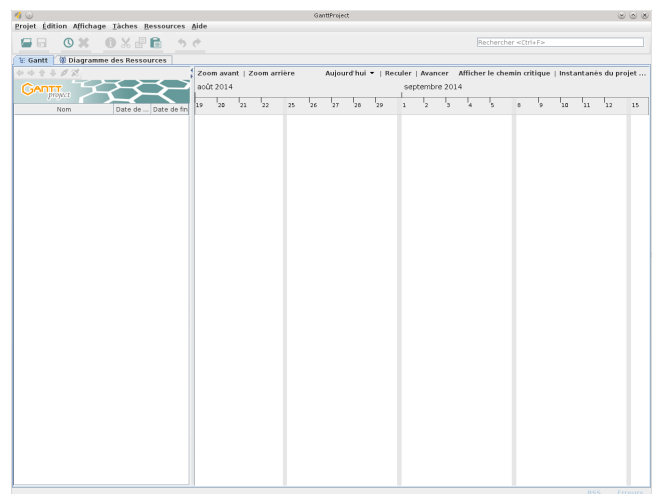
L'installation est simple et je ne vais pas la détailler ici :

- sous Windows, il vous suffit de suivre les étapes ;
- sous Linux, il vous suffit de récupérer le paquet correspondant.

Une fois l'installation faite, voici ce que vous obtenez, dès que le programme est lancé :



sous Windows



sous Linux (OpenSuse)

Comme vous pouvez le constater, les interfaces sont légèrement différentes suivant l'environnement, mais les fonctionnalités restent les mêmes. Les commandes seront alors visibles sous forme de texte ou dans la barre d'outils.

## 3 Prérequis

Le point de départ de tout projet va être de devoir déterminer les différents besoins en termes :

- de décomposition des tâches ;
- d'affectation des ressources ;

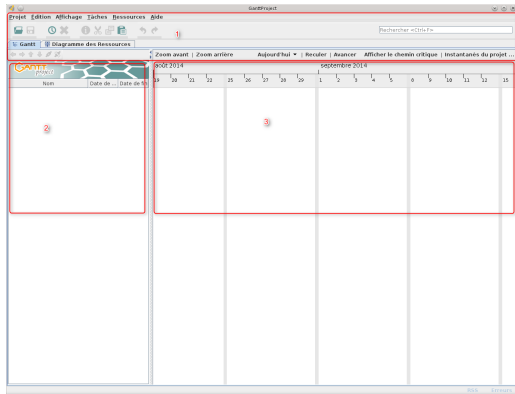
— de délais.

Et avant toute chose, nous allons devoir déterminer les éléments suivants pour chaque tâche :

- sa durée ;
- sa date de début ou de fin ;

## 4 Interface

L'interface se présente donc sous la forme suivante :



- 1 : contient toutes les commandes ;
- 2 : tableau listant les différentes tâches ;

— la ou les ressources ;

— le temps passé par jour pour chaque ressource ;

— la ou les tâches précédentes ;

— la ou les tâches suivantes.

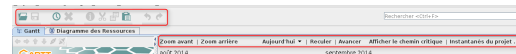
— 3 : donne une représentation du diagramme de Gantt.

Nous pouvons voir aussi sur ces images qu'il y a deux onglets, le premier donne le diagramme de Gantt du projet, et le second servira à fournir les informations sur l'utilisation des ressources. Un onglet pourra apparaître, il s'agit « Diagramme de PERT », mais nous l'aborderons dans le paragraphe 9.2.

Comme je l'ai indiqué précédemment, les commandes ne sont pas toujours au même endroit, mais elles y sont toutes :



sous Windows

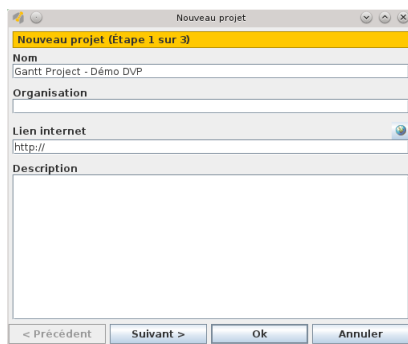


sous Linux (OpenSuse)

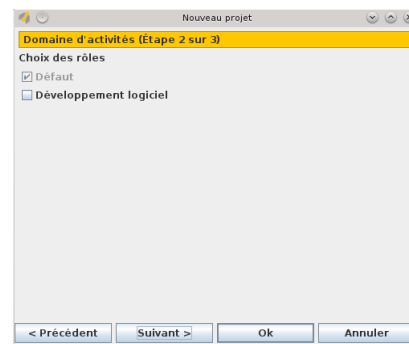
## 5 Création d'un projet

En ouvrant GanttProject, nous arrivons sur un projet nouveau. Si nous utilisons la commande « Nouveau » du menu « Projet », nous allons utiliser l'assistant qui se décompose en trois étapes.

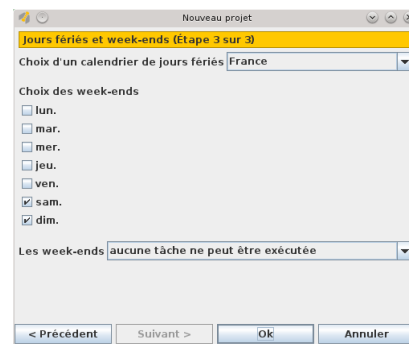
La première permet de définir le paramètre général du projet :



La deuxième permet de définir le domaine d'activité :



La dernière concerne le calendrier (pour les jours fériés) et les jours ouvrés :



Une fois que ces étapes sont renseignées, vous pouvez créer votre projet en cliquant sur « OK » ; n'oubliez pas de l'enregistrer.

## 6 Exemple

Maintenant que nous allons passer à la construction du projet dans le logiciel, il faut que les différentes tâches soient connues.

Pour notre exemple, prenons le cas suivant :

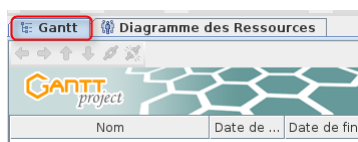
Tâches	Description	Prédécesseurs	Durée (jour)
A	Libellé A		3
B	Libellé B	A	2
C	Libellé C		10
D	Libellé D	B, C	1
E	Libellé E	C	5
F	Libellé F	D, E	8
G	Libellé G	F	10



Et il est très important de bien définir toutes les tâches, cela n'en sera que plus facile pour utiliser et renseigner le programme par la suite.

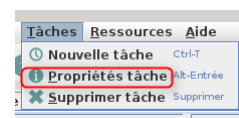
## 7 Les tâches

Les tâches se trouvent dans l'onglet Gantt :



Maintenant, pour compléter correctement les informations, il faut se mettre sur la tâche et activer les propriétés de la tâche :

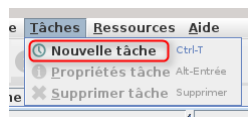
— avec le menu :



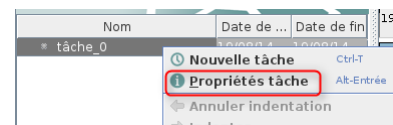
### 7.1 Créer une tâche

Pour créer une tâche, la commande se trouve :

— dans le menu :



— dans la fenêtre des tâches en faisant un clic droit :



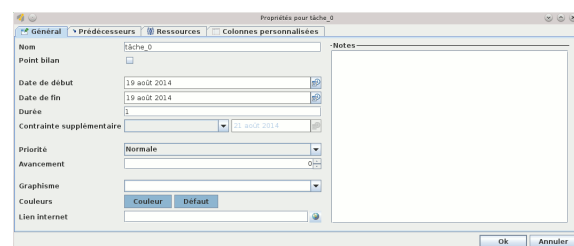
— dans la barre d'outils :



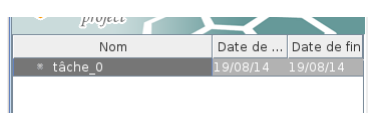
— dans la fenêtre des tâches en faisant un clic droit :



Une fois la commande sélectionnée, la fenêtre suivante s'ouvre :

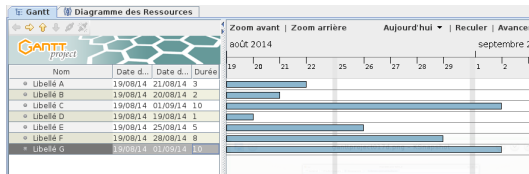


Une fois la demande faite, une nouvelle ligne apparaît dans la liste :



Sur cette fenêtre, il y a beaucoup d'informations qui sont réparties dans quatre onglets (ou cinq sous Windows). Toutes ces informations ne sont pas à saisir au moment de la création, mais elles pourront être complétées ou modifiées au cours de la gestion et de l'organisation du projet.

Dans notre cas et avec les éléments fournis dans le tableau précédent, je n'ai pour le moment saisi que le nom des tâches et la durée, nous allons ainsi obtenir la liste suivante :

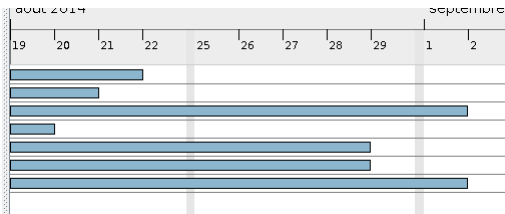
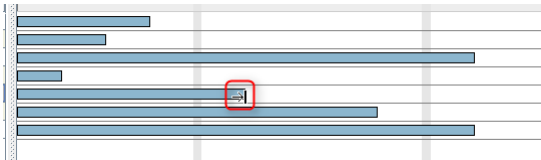


Dans la partie gauche, nous avons les tâches, et dans la partie droite la représentation de la durée.

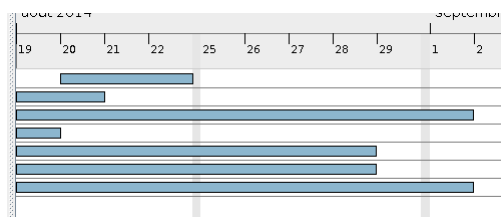
## 7.2 Gérer et organiser les tâches

Passons maintenant à la partie gestion et organisation des tâches. Nous savons que toutes les données sont modifiables avec la fenêtre des propriétés, mais certaines le sont sur le graphique :

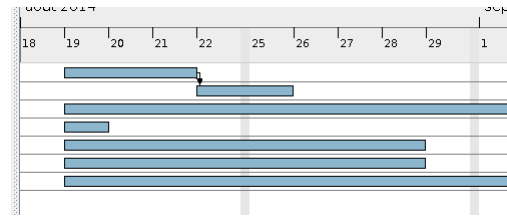
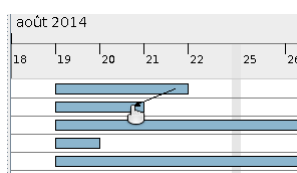
- la durée peut être modifiée, il suffit pour cela de se placer sur le bord du segment et de faire glisser en moins ou en plus :



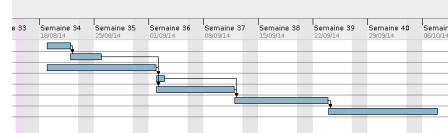
- les dates sont aussi modifiables sur le graphique, pour cela, il nous suffit de maintenir la touche CTRL enfoncée et avec le clic droit de déplacer le segment :



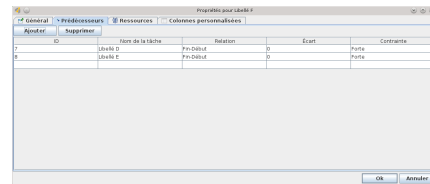
- la définition des précédents est aussi réalisable dans le graphique, il nous suffit de cliquer le segment de départ et de le faire glisser vers le segment d'arrivée :



La représentation finale de notre exemple donne donc :

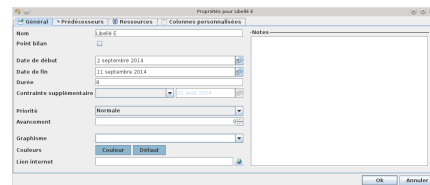


Mettons-nous sur la tâche « Libellé F » et activons la fenêtre des propriétés dans l'onglet « Précédenceurs » :

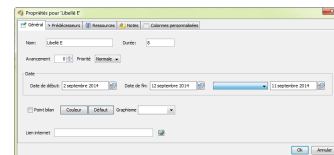


## 7.3 La fenêtre propriété

Voici une représentation de l'onglet « Général » :



sous Linux (OpenSuse)



sous Windows



Comme vous pouvez le voir, les éléments n'ont pas la même disposition, mais ceux-ci sont présents dans la fenêtre.

### 7.3.a Onglet « Général »

#### Nom

Celui-ci peut-être modifié tout au long de la vie du projet. Il est préférable de bien expliquer à quoi correspond la tâche, cela n'en sera que plus facile pour la lecture globale du projet.

#### Point bilan

Il permet un événement sur une journée, il est représenté ainsi dans le graphique :



Très utile pour symboliser une réunion !

### Dates et durée

Ces trois éléments sont liés, il suffit d'en renseigner un pour que les deux autres se mettent à jour.



Si vous modifiez une des deux dates, c'est la durée qui évolue.

### Contrainte supplémentaire

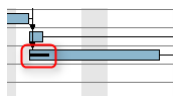
Et une information complémentaire sur la date minimum de début d'une tâche.

### Priorité

Cela permet de définir une priorité sur une tâche.

### Avancement

Il suffit de mettre une valeur allant de 0 à 100 sur l'état d'avancement de la tâche, une fois renseignée, cela sera visible sur le graphique :

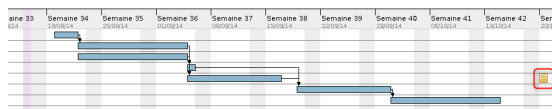


### Graphisme et couleurs

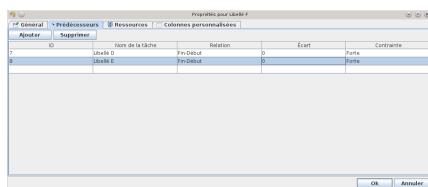
Cela permet de personnaliser le segment, en y apportant un motif et une couleur. Cela peut être utile si vous souhaitez différencier les tâches de préparations, de tests, de déploiements, etc.

**Notes** (*sous Windows se trouve dans un onglet à part*)

Cette zone permet de saisir du texte, ce qui peut être pratique pour faire le suivi du projet.



### 7.3.b Onglet « Prédécesseurs »



Sur cet onglet, nous allons pouvoir identifier les tâches qui devront être réalisées pour exécuter la tâche sélectionnée.

### Nom de la tâche

Il suffit pour cela de sélectionner à l'aide de la liste déroulante la ou les tâches qui devront les précéder. Cela mettra à jour automatiquement les dates de début et de fin.

### Relation

Normalement, une tâche ne peut commencer tant que l'autre n'est pas finie, mais il existe d'autres relations.

**Fin-Début** : c'est la plus couramment utilisée.

Une tâche ne peut commencer si la précédente n'est pas terminée.

**Fin-Fin** : dans ce cas-là, les deux tâches se termineront en même temps.

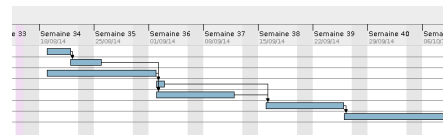
**Début-Début** : dans ce cas-là, les deux tâches commenceront en même temps.

**Début-Fin** : dans ce cas-là, la tâche ne pourra commencer qu'au moment où l'autre sera terminée.

### Écart

Cette valeur permet d'ajouter une durée à la tâche sans que celle-ci ne soit comprise dans son déroulement.

Par exemple, sur la tâche F, si nous mettons 2 sur la tâche E, voici ce que devient le graphique :



### Contrainte

Cette fonctionnalité permet de définir la forme du trait sur le graphique :

- forte : trait continu ;
- faible : trait pointillé.

### 7.3.c Onglet « Ressources »

Cet onglet sera abordé dans le paragraphe des ressources.

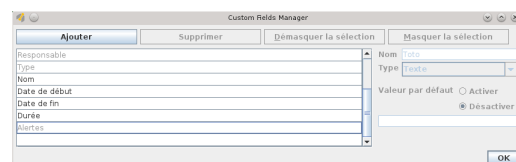
### 7.3.d Colonnes personnalisées

Cet onglet n'est pas propre à la tâche. En effet, il permet de sélectionner les éléments que nous souhaitons voir apparaître dans la liste des tâches.

Cette commande est aussi accessible en faisant un clic droit sur le titre des colonnes :



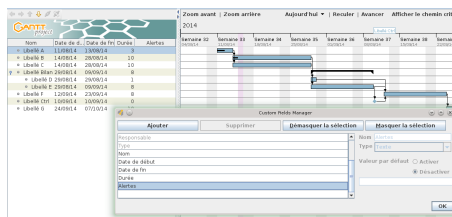
La fenêtre de gestion des colonnes est la suivante :



Le texte en noir correspond aux colonnes visibles, le texte en gris correspond aux colonnes que nous pouvons activer, il suffit pour cela de sélectionner la colonne et de cliquer sur le bouton « Démasquer la sélection » :



La colonne apparaît dans la liste :



Vous pouvez aussi ajouter de nouvelles colonnes, celles-ci seront alimentées manuellement par la suite, pour cela il suffit de cliquer sur « Ajouter » :



Ensuite, il nous faut nommer la colonne, ce qui permettra d'activer la partie permettant de définir le type de donnée et la valeur par défaut s'il y en a une.

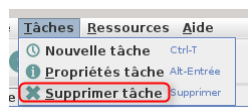
La colonne apparaît ensuite dans la liste, il ne restera plus qu'à la remplir manuellement.

## 7.4 Suppression/Insertion

### 7.4.a Supprimer une tâche

Pour supprimer une tâche, il faut la sélectionner ensuite :

— dans le menu :



— dans la barre d'outils :

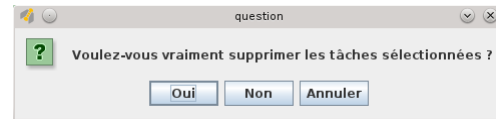



— dans la fenêtre des tâches avec un clic droit :



— avec le clavier et la touche « Suppr ».

Une fenêtre de confirmation apparaîtra :



 Vous pouvez en supprimer plusieurs à la fois, il faut pour cela toutes les sélectionner avec Ctrl et ensuite supprimer.

### 7.4.b Insérer une tâche

Pour insérer une tâche, cela fonctionne comme pour la création (voir le paragraphe de création), la nouvelle tâche sera insérée à la suite de la tâche active :

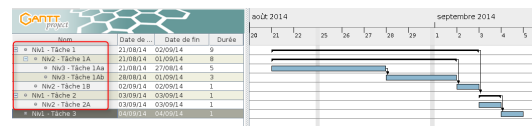
Libellé E	02/09/14	11/09/14	8
tâche 11	11/08/14	11/08/14	1
Libellé F	16/09/14	25/09/14	8
Libellé G	26/09/14	09/10/14	10

Il ne vous restera plus qu'à compléter ses informations, etc.

## 7.5 Arbre hiérarchique ou sous-tâche

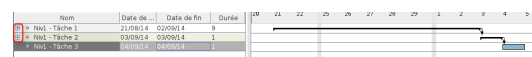
Si vous avez beaucoup de tâches à mettre dans votre projet, cela devient rapidement illisible, vous avez donc la possibilité de créer des arbres hiérarchiques ou sous-tâches.

Voici un exemple d'arbre hiérarchique et de sa représentation graphique :



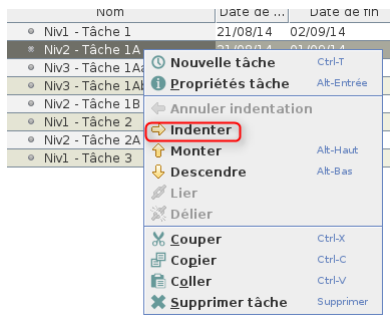
Dans la représentation graphique, les arbres ne sont plus représentés par des segments, mais par des traits, ceux-ci tiennent compte des éléments qui sont inclus dans les niveaux inférieurs.

Cette utilisation permet de simplifier les affichages :

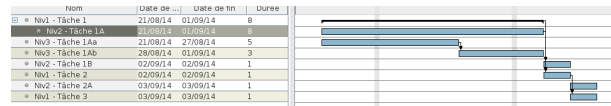



Pour créer un arbre hiérarchique, il faut se mettre sur la tâche et faire un clic droit, puis sélectionner la commande « Indenter » :





L'arbre sera automatiquement créé :



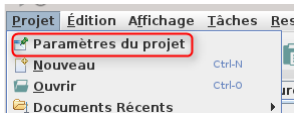
 Au même niveau que la commande « Indenter », il y a la commande pour l'annuler.

## 8 Les ressources

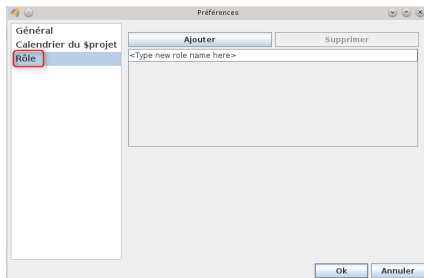
### 8.1 Prérequis

Avant de créer toutes les ressources, il est nécessaire de définir tous les rôles des ressources (nous pourrions y retourner par la suite), car cette information nous sera demandée.

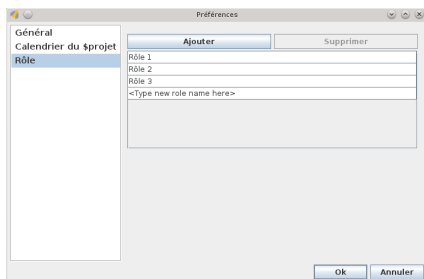
La commande se trouve dans le menu « Projet » et « Paramètres du projet » :




La fenêtre suivante s'ouvre, il suffit alors de sélectionner la commande « Rôle » :

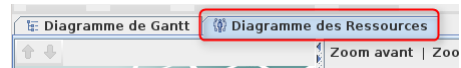


Il ne nous reste plus qu'à ajouter les différents rôles qui seront nécessaires dans le projet :

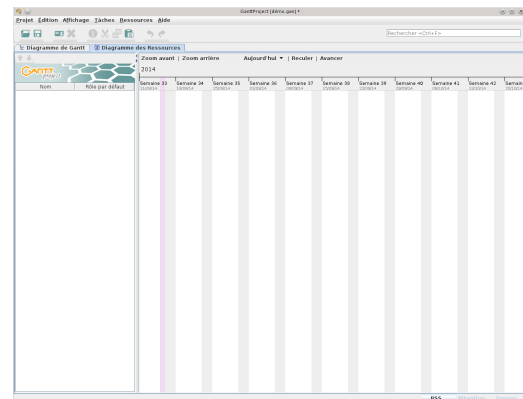


 Il est inutile de définir le rôle « Chef de projet », celui-ci étant par défaut dans la liste.

Ensuite, il suffit de se rendre dans l'onglet des ressources :



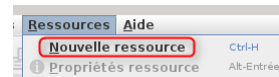
La fenêtre devient donc :



### 8.2 Créer les ressources

Pour créer une ressource, la commande se trouve :

— dans le menu :



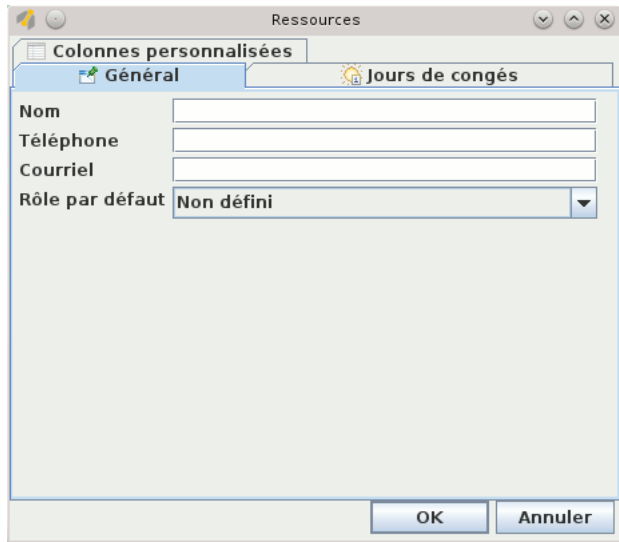
— dans la barre d'outils :



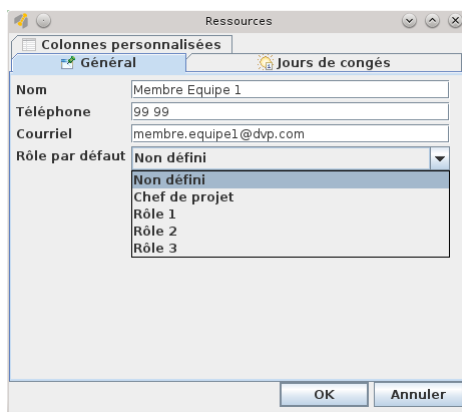
— dans la fenêtre des tâches en faisant un clic droit :



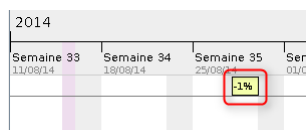
La fenêtre suivante s'ouvre :



Dans le premier onglet, il nous reste à saisir le nom et les éléments dont vous disposez :



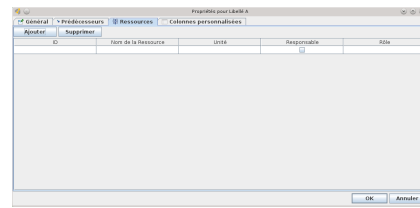
Vous pourrez à tout instant rajouter des absences ou des indisponibilités dans l'onglet « Jours de congés ». Il faudra saisir la date de début et la date de fin soit manuellement, soit avec le calendrier. La représentation sur le graphique sera la suivante :



Le troisième onglet « Colonnes personnalisées » n'est pas propre à la ressource. En effet, il permet de sélectionner les éléments que nous souhaitons voir apparaître dans la liste des ressources.

### 8.3 Affecter les ressources

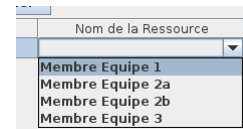
L'affectation des ressources sur une tâche se fait dans la fenêtre des propriétés d'une tâche :



Dans cette fenêtre, il suffit ensuite de sélectionner les différents éléments nécessaires.

#### Nom de la Ressource

À l'aide de la liste déroulante, il suffit de choisir la ressource :



La première personne sélectionnée sera automatiquement définie comme la personne responsable.

#### Unité

Cette information correspond au pourcentage de temps de travail que la ressource doit passer sur cette tâche.

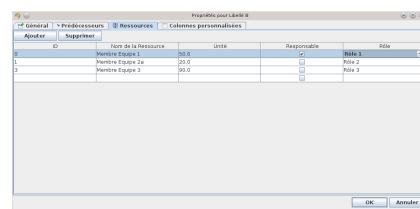
#### Responsable

Cette case à cocher sert à définir qui sera responsable de cette tâche.

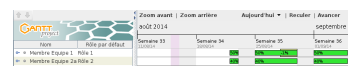
#### Rôle

Si la ressource a bien été remplie comme il faut, l'information remonte automatiquement une fois le nom sélectionné.

Sur la tâche « Libellé B », voici ce que cela peut donner :



Ensuite, nous procédons de la même façon pour la tâche « Libellé C ». Par la suite, pour voir l'occupation des ressources, il suffit de retourner sur l'onglet « Diagramme des ressources », ce qui nous donne :



Nous pouvons voir la répartition en cliquant sur le petit sigle à gauche de la ressource :



Nous obtenons ainsi la décomposition de la charge de la ressource en fonction des tâches pour lesquelles elle est sollicitée.

Maintenant, si vous avez affecté trop à une ressource, la couleur « Verte » sera alors remplacée par la couleur « Rouge » :



Il ne vous restera plus qu'à corriger les éléments nécessaires. Vous pouvez modifier directement en

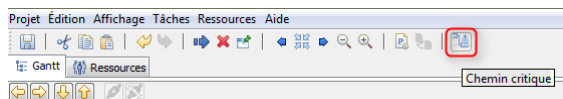
## 9 Suivi du projet

GanttProject est un outil permettant de faire le suivi de vos projets. Il dispose pour cela de différents outils pour vous aider dans l'analyse et la diffusion des documents.

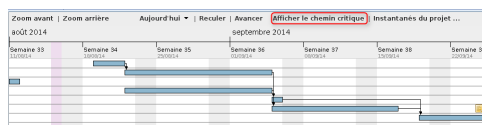
### 9.1 Chemin critique

Le chemin critique désigne la liste ordonnée des opérations à réaliser pour obtenir le résultat demandé, sa durée donne la durée du projet.

La commande se trouve :

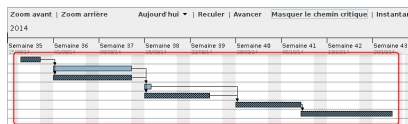


sous Windows



sous Linux (OpenSuse)

Nous obtenons ensuite le graphique suivant :



Les segments hachurés représentent le chemin critique.



Pour faire disparaître les informations du chemin critique, il suffit de cliquer sur la commande qui l'a fait apparaître.

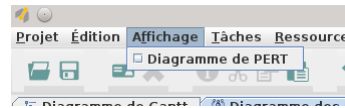
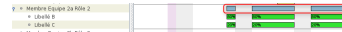
### 9.2 PERT

Ce terme signifie « Project Evaluation and Review Technique ». Cette méthode conventionnelle est utilisée en gestion de projet, cela permet de représenter et d'analyser de manière logique les tâches du projet.

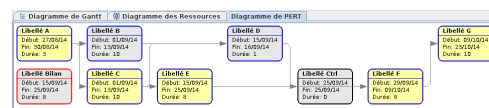
La commande se trouve dans le menu « Affichage » :

faisant un clic droit sur le nom de la tâche ou revenir dans l'onglet correspondant.

Si la ressource a une charge égale à 100, la couleur « Verte » sera alors remplacée par la couleur « Bleue », et l'information en pourcentage ne s'affiche pas :



Nous obtenons un nouvel onglet « Diagramme de PERT » :



Chaque couleur a sa signification :

<p><b>Libellé Bilan</b> Début: 15/09/14 Fin: 25/09/14 Durée: 8</p>	La bordure rouge permet d'identifier une tâche qui est composée de sous-tâches.
<p><b>Libellé A</b> Début: 27/08/14 Fin: 30/08/14 Durée: 3</p>	Le fond jaune représente une tâche critique.
<p><b>Libellé B</b> Début: 01/09/14 Fin: 13/09/14 Durée: 10</p>	Cela correspondant à une tâche normale.
<p><b>Libellé Ctrl</b> Début: 25/09/14 Fin: 25/09/14 Durée: 0</p>	La bordure noire correspond à un point bilan.





Pour faire disparaître l'onglet PERT, il suffit de cliquer sur la commande qui l'a fait apparaître.

### 9.3 Alertes et retards

L'outil permet aussi d'identifier les retards sur le planning, en fonction de l'état d'avancement de la tâche, de la date de fin et de la date du jour.

Pour que cette information apparaisse, la colonne « Alerte » doit être active dans la liste des tâches, et nous obtenons donc :

Nom	Date de dé.	Date de fin	Durée	Alertes
Libellé A	11/08/14	13/08/14	3	
Libellé B	14/08/14	28/08/14	10	
Libellé C	14/08/14	28/08/14	10	
Libellé Bilan	15/09/14	25/09/14	8	

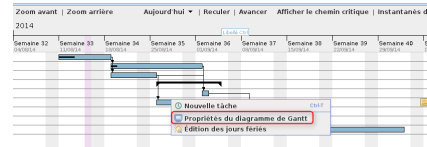
	Indique un retard sur la tâche.
	Indique un retard sur la tâche, mais celle-ci est sur le chemin critique.

Nous constatons que le graphique évolue, les « petits » segments représentent le projet sauvegardé, les couleurs correspondent au projet en cours.

### 9.5 Personnalisation du graphique

Les graphiques contiennent certaines informations par défaut. Mais il est possible de les modifier, d'en rajouter et d'en supprimer.

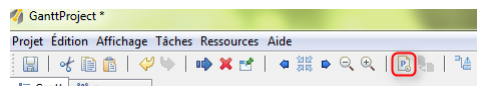
La commande se trouve sur le clic droit sur le graphique :



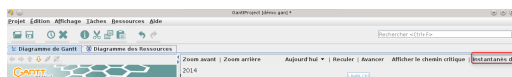
### 9.4 Enregistrement d'un état

Cette fonctionnalité permet à tout moment d'enregistrer l'état du projet, celui-ci pouvant être comparé par la suite au projet en cours.

La commande se trouve :



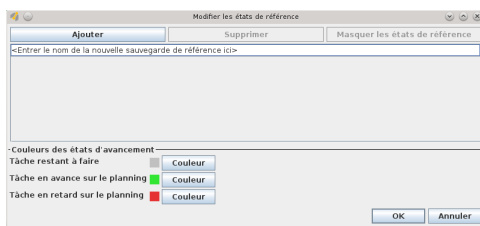
sous Windows



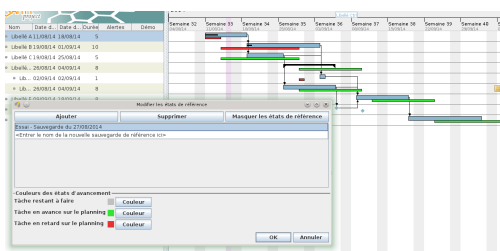
sous Linux (OpenSuse)

 La commande se trouve aussi dans le menu « Édition/Préférences ».

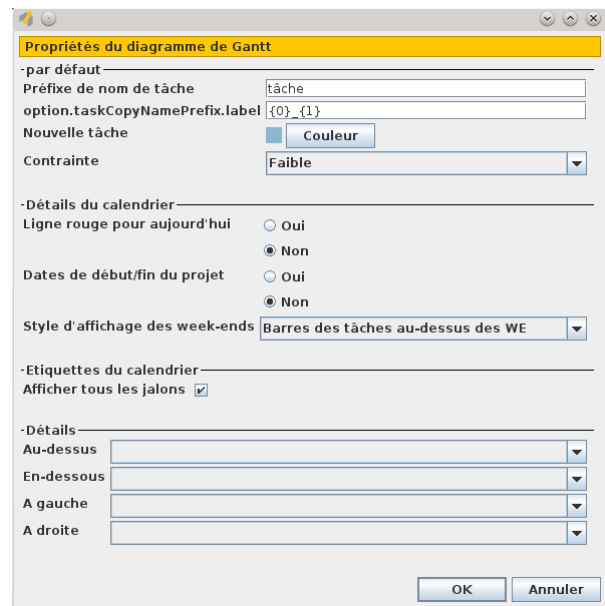
La fenêtre suivante s'ouvre :



Il nous suffit de nommer la sauvegarde. Ensuite, le projet va vivre et les éléments vont bouger. Si nous voulons comparer l'état actuel du projet avec une version sauvegardée, il suffit de la sélectionner :



La fenêtre qui s'ouvre et la suivante :



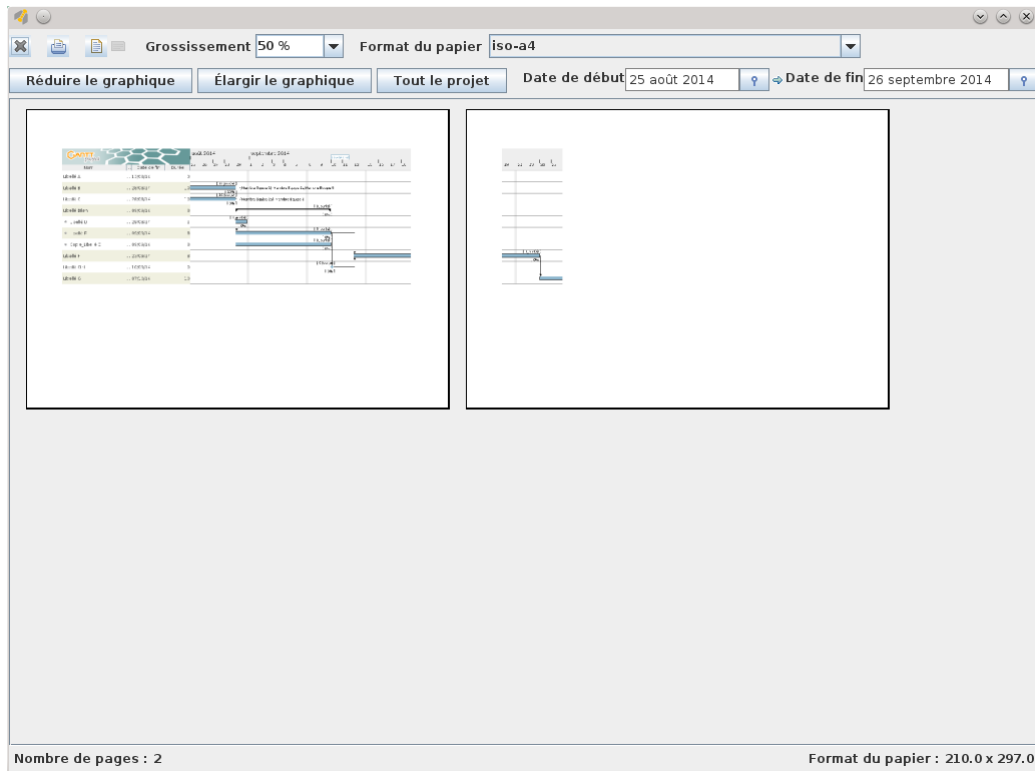


### 9.6.a Imprimer

L'impression se fait par défaut sur papier de format A4 et imprime l'ensemble du projet correspondant à ce qui est visible à l'écran.

### 9.6.b Aperçu avant impression

Par défaut, l'aperçu correspond à celui qui serait imprimé :

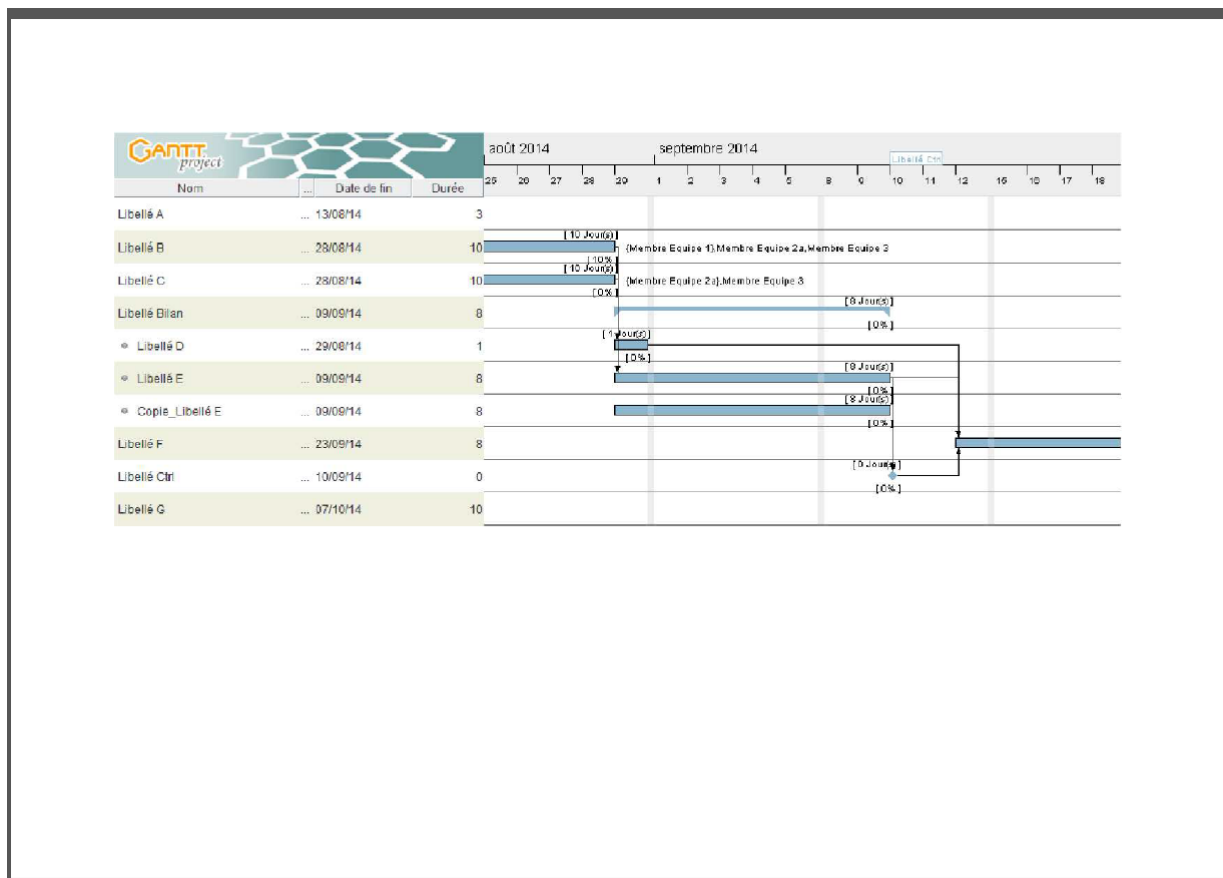


Sur cette fenêtre, nous avons la possibilité de modifier les paramètres suivants :




	<p>Permet de choisir l'orientation de la page, la partie grisée indique la position en cours.</p> <p>Permet avec l'aide de la liste déroulante d'agrandir ou réduire l'affichage dans la fenêtre d'aperçu.</p> <p>Permet de choisir le format de papier.</p> <p>Permet de modifier l'échelle du temps sur le graphique.</p> <p>Permet de modifier la zone à imprimer en modifiant les dates de début et de fin, qui correspondent à la plage que vous souhaitez imprimer.</p> <p>Permet de remettre la plage d'impression à la totalité du projet.</p> <p>Permet d'imprimer ce que vous avez sélectionné.</p> <p>Cette information se trouve en bas à gauche de la fenêtre, elle vous indique en un coup d'œil le nombre de pages qui seront imprimées.</p>
--	---

Ce qui nous donne donc finalement quelque chose comme ceci :



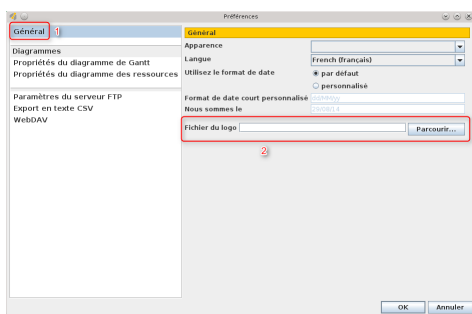
## 9.7 Modifier le logo

Comme nous venons de le voir dans l'impression faite dans le paragraphe précédent, le logo par défaut est celui de GanttProject.

 La modification du logo se fera pour application sur tout le projet.


Pour cela, vous devez disposer d'une image au format png, l'image ne devant pas dépasser 47 pixels de haut.

La commande se trouve dans le menu « Édition/Préférences », dans la partie « Général », vous trouverez la commande permettant de choisir le fichier du logo :

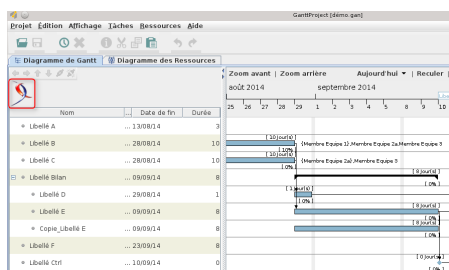


En cliquant sur « Parcourir », il ne vous reste plus qu'à aller chercher votre logo, et finalement nous obtenons :



 Parfois, il est nécessaire de redémarrer l'application pour que le nouveau logo soit pris en compte.

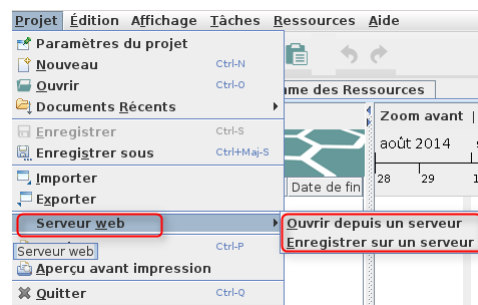
Le logo enregistré sera maintenant présent sur toutes les impressions, mais il est aussi présent maintenant à l'affichage :



## 9.8 Serveur

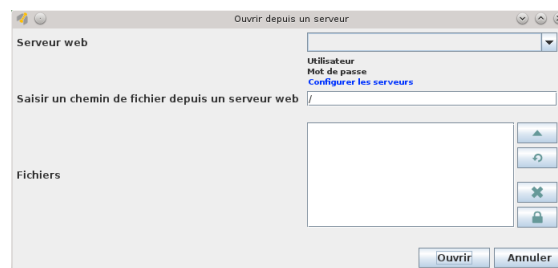
GanttProject permet de mettre votre fichier Web sur serveur, mais il faut pour cela un serveur WebDAV.

La commande se trouve dans le menu « Projet » :



Vous avez la possibilité d'ouvrir un projet ou d'enregistrer votre projet sur le serveur.

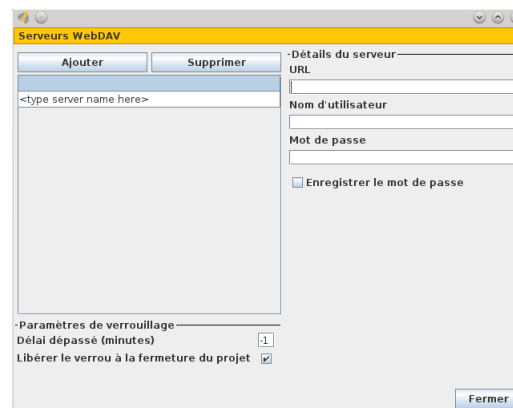
Les fenêtres qui apparaissent à l'ouverture et à l'enregistrement sont très similaires, seul un bouton change :



Pour créer un fichier : il vous suffira de sélectionner le serveur, donner un nom à votre fichier « nom.xml » et cliquer sur « Enregistrer ».

Pour ouvrir un fichier : il vous suffira de sélectionner le serveur, indiquer le chemin, actualiser, sélectionner le fichier et cliquer sur « Ouvrir ».

Pour définir un serveur, il ne faut cliquer que sur « Configurer les serveurs », la fenêtre suivante s'ouvre :



Il suffit de saisir les différents éléments demandés. Par suite, le serveur apparaîtra dans la liste déroulante.

## 9.9 Importation/exportation

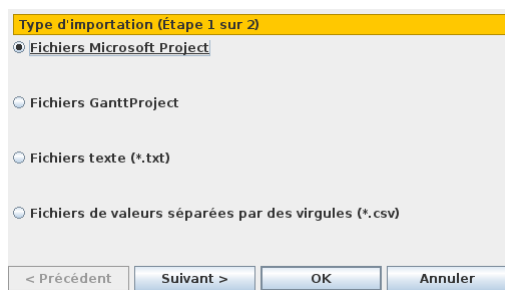
### 9.9.a Importation

Vous pouvez importer des informations provenant d'autres applications ou de certains fichiers, vous évitant ainsi une ressaisie.

La commande se trouve dans le menu « Projet » :



La fenêtre qui apparaît est la suivante :



Dans les cas suivants : Fichiers Microsoft Project, Fichiers textes, Fichiers CSV, il vous suffit de sélectionner le nom du fichier.

Pour les fichiers GanttProject, il faut sélectionner le fichier, mais il vous est aussi demandé une information pour fusionner les ressources en double : par le nom, l'ID ou le mail, ou de ne pas fusionner les ressources.

Les éléments importés pourront être ajoutés au projet en cours à la suite des autres ou dans un nouveau projet.

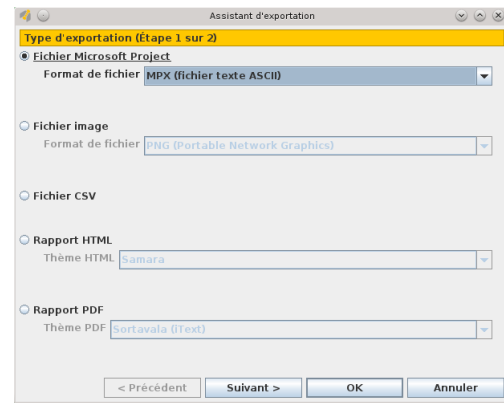
### 9.9.b Exportation

Si vous souhaitez diffuser l'information de votre projet, Ganttproject vous permet d'exporter les différents éléments dans divers formats : PDF, HTML, image et même vers Microsoft Project.

La commande se trouve dans le menu « Projet » :



La fenêtre qui apparaît est la suivante :



Maintenant, au niveau de l'exportation que vous souhaitez réaliser, les informations demandées ne sont pas les mêmes.

#### 9.9.b.a Microsoft Project

Comme vous pouvez le constater, il y a deux types de données possibles :

- MPX : pour les versions allant jusqu'à MS Project 98 ;
- MSPDI : pour les versions récentes de MS Project.

Les différences viennent de la construction du fichier, le MSPDI est un format XML.

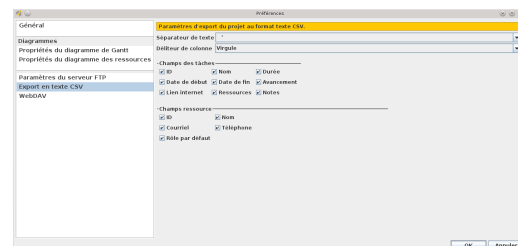
Les informations demandées sont l'emplacement du fichier, la publication sur FTP et, dans le cas d'un MPX, la langue.

#### 9.9.b.b Image

Vous pouvez enregistrer au format jpg ou png. Les informations demandées sont l'emplacement du fichier, la date de début et de fin (par défaut le projet entier).

#### 9.9.b.c CSV

Vous pouvez enregistrer un fichier au format CSV, les données qui seront exportées sont paramétrables dans le menu « Édition/Préférences » et dans la commande « Export en CSV ». Il vous est possible de sélectionner le séparateur, le délimiteur de texte et les données à exporter :



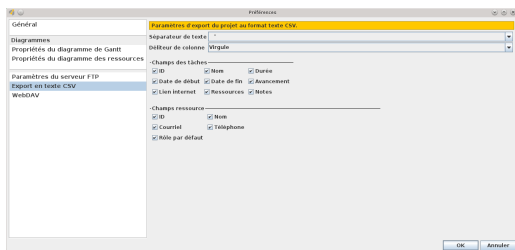
Les informations demandées sont l'emplacement du fichier, la date de début et de fin (par défaut le projet entier).

### 9.9.b.d HTML

Cette commande vous permet d'exporter les informations en plusieurs pages Web. Ces données sont :

- les informations générales du projet ;
- les graphiques de tâches et ressources ;
- la liste des tâches avec les informations complètes (dates, ressources, etc.) ;
- la liste des ressources.

Pour cela, GanttProject propose des thèmes :



Mais cela pourra être modifié avec une feuille de style et en modifiant le code HTML des pages. Sinon, vous pouvez aussi vous créer des modèles en composant des feuilles de styles XSL, ces données sont contenues dans le répertoire de GanttProject :

Les informations demandées sont l'emplacement du fichier, la date de début et de fin (par défaut le projet entier).

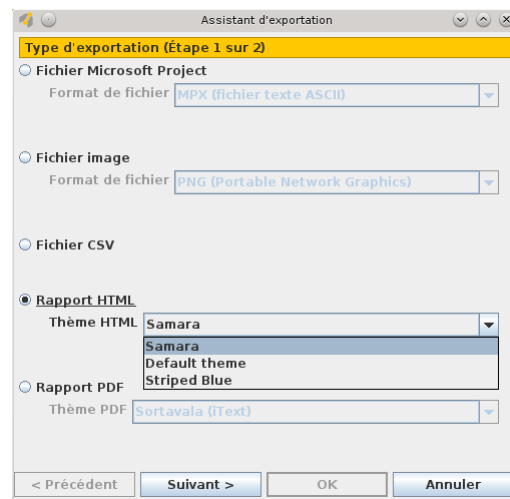
### 9.9.b.e PDF

Cette commande vous permet d'exporter un fichier PDF contenant :

- les informations générales du projet ;
- les graphiques tâches et ressources ;
- la liste des tâches avec les informations complètes (dates, ressources, etc.) ;
- la liste des ressources.

Pour cela, GanttProject propose des thèmes :

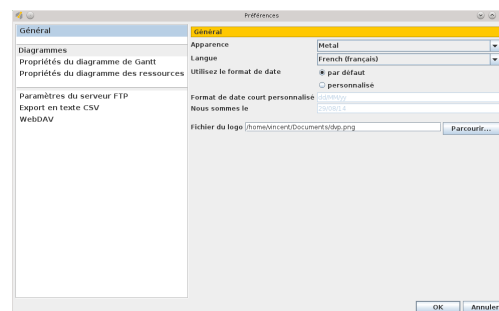
Retrouvez l'article de **Vincent Viale** en ligne : [lien 60](#)



Mais cela pourra être modifié avec une feuille de style et en modifiant le code HTML des pages. Sinon, vous pouvez aussi vous créer des modèles en composant des feuilles de styles XSL, ces données sont contenues dans le répertoire de GanttProject : /plugins/org.ganttproject.impex.htmlpdf/resource/ Les informations demandées sont l'emplacement du fichier, la date de début et de fin (par défaut le projet entier).

### 9.10 Complément

La fenêtre « Préférences » qui se trouve dans le menu « Édition » permet de modifier beaucoup de choses :



**!** Tous les éléments modifiés dans cette fenêtre impactent l'application et pas le projet en cours.

Vous pouvez modifier les apparences de l'application, les serveurs, etc.

# Apprendre à utiliser ProjectLibre

Au cours de ce tutoriel, je ne vais pas vous apprendre à gérer un projet, mais je vais vous montrer comment utiliser ProjectLibre qui est un des outils libres de gestion de projets.

## 1 Définition

ProjectLibre est un logiciel libre, permettant de modéliser sous forme de Gantt la planification des différentes tâches qui sont nécessaires à un projet. Chaque tâche pourra être définie par sa durée, par les ressources qui pourront lui être affectées et par des budgets. Vous pourrez aussi obtenir des documents comme le PERT et le chemin critique.

Il est une reprise du projet abandonné OpenProj, qui se voulait un substitut libre à MS Project. Ce qui fait que ProjectLibre a une interface similaire, et qu'il est aussi possible d'ouvrir les fichiers de MS Project.

## 2 Installation

C'est un logiciel multiplateforme (Linux, Mac OS et Windows). La page de téléchargement se trouve sur le site de ProjectLibre : [lien 61](#). Si vous êtes sous Windows ou Linux, vous pourrez trouver une version portable de l'outil.

L'installation est simple et je ne vais pas la détailler ici :

- sous Windows, il vous suffit de suivre les étapes ;
- sous Linux, il vous suffit de récupérer le paquet correspondant.

Une fois l'installation faite, voici ce que vous obtenez, dès que le programme est lancé :

## 3 Prérequis

Le point de départ de tout projet va être de devoir déterminer les différents besoins en termes :

- de décomposition des tâches ;
- d'affectation des ressources ;
- de délais.

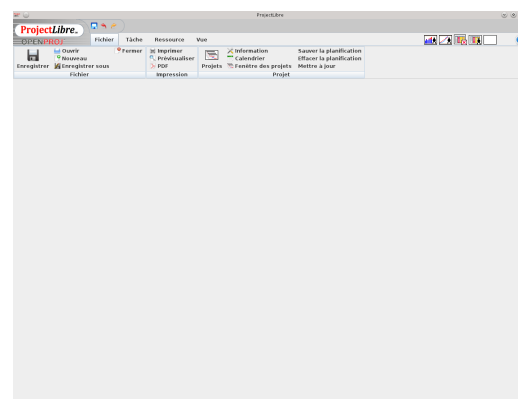
Et avant toute chose, nous allons devoir déterminer les éléments suivants pour chaque tâche :

## 4 Interface

L'interface se présente donc sous la forme suivante, une fois que vous êtes dans un projet :

Les logiciels de gestion de projets sont généralement utilisés pour automatiser la sauvegarde des tâches et la gestion du temps. La visualisation d'un Gantt permet, par un simple schéma graphique, de savoir qui fait quoi en donnant l'état d'avancement du projet, de chaque tâche et sous-tâche, d'avoir la planification de l'ensemble et la charge de travail des équipes.

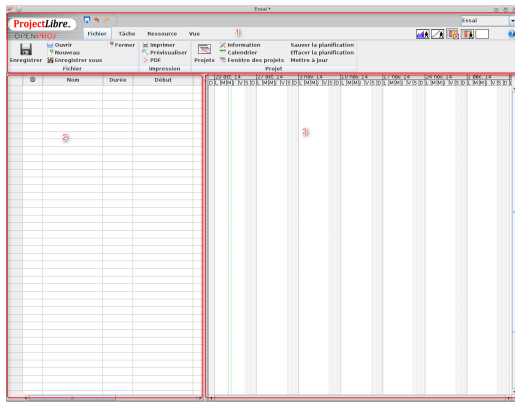
Les éléments obtenus pourront être imprimés ou convertis en PDF, vous aurez aussi la possibilité d'exporter les données vers MS Project.



Les interfaces sont identiques, quel que soit l'environnement.

- sa durée ;
- sa date de début ou de fin ;
- la ou les ressources ;
- le temps passé par jour pour chaque ressource ;
- la ou les tâches précédentes ;
- la ou les tâches suivantes.





— 1 : contient toutes les commandes ;

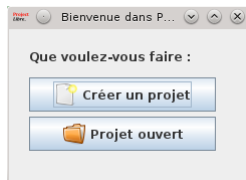
- 2 : tableau listant les différentes tâches ;
- 3 : donne une représentation du diagramme de Gantt.

Nous pouvons voir aussi sur cette image qu'il y a différentes commandes :

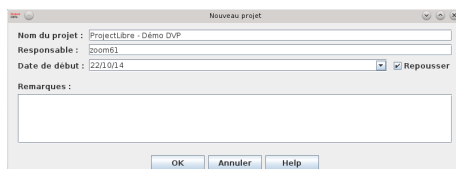
- « Fichier » qui nous permettra de gérer et modifier les informations du projet ;
- « Tâche » qui permet de gérer, modifier et supprimer les tâches ;
- « Ressource » qui permet de gérer, modifier et supprimer les ressources du projet ;
- « Vue » qui permet de regrouper toutes les commandes de visualisation du projet.

## 5 Création d'un projet

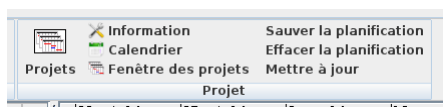
En ouvrant ProjectLibre, nous arrivons sur ce masque :



En cliquant sur « Créer un projet », nous obtenons :

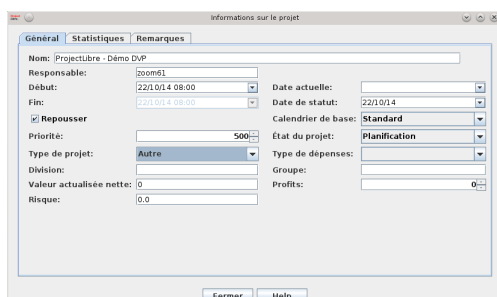


Les informations relatives au projet se trouvent dans ce menu :

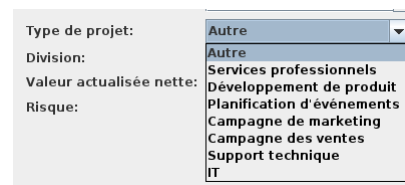


### 5.1 Les informations générales

La commande « Information » ouvre la fenêtre suivante :

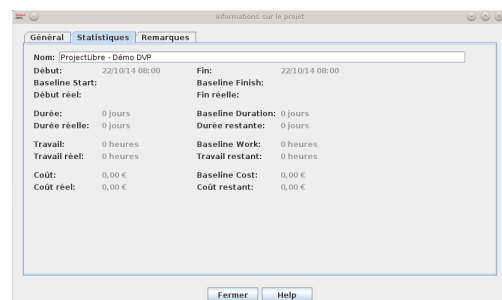


Le premier onglet contient les données modifiables du projet comme le type de projet :

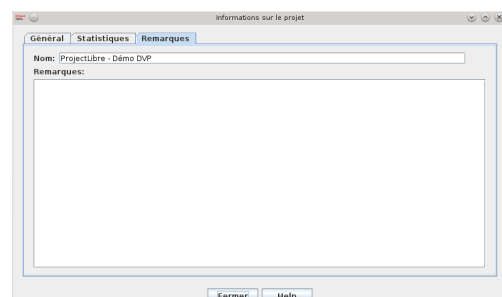


Mais nous disposons d'autres données modifiables comme le type de calendrier, l'état du projet, etc.

Le deuxième onglet « Statistiques » contient toute la synthèse du projet en compilant toutes les données :



Le troisième onglet « Remarques » permet de saisir des éléments textuels sur le projet :

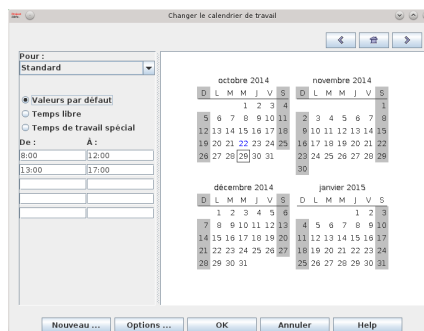


## 5.2 Le calendrier

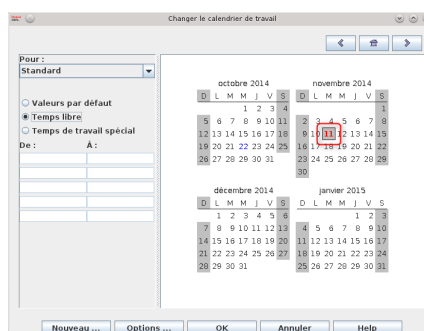
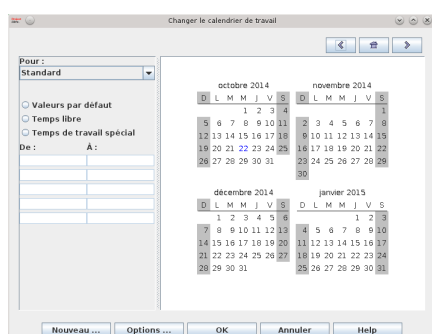
Par défaut, si vous avez sélectionné le calendrier « Standard », la plage de travail sera la suivante (ces données pourront être modifiées par la suite) :

- le samedi et le dimanche sont des jours fermés ;
- la plage horaire pour les autres jours est 8 h-12 h et 13 h-17 h.

La commande « Calendrier » ouvre la fenêtre suivante :

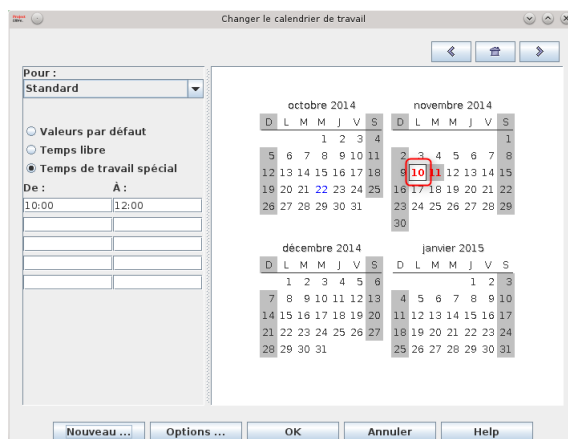


Une fois le jour sélectionné, si vous cochez « Temps libre », cela signifiera que le jour n'est pas travaillé, et il deviendra grisé et rouge sur le calendrier :

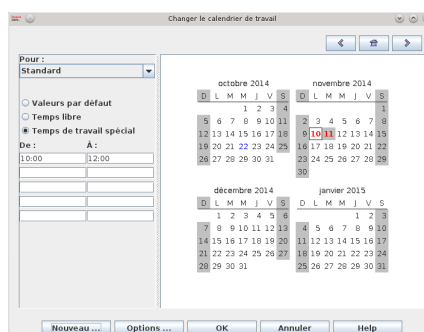


Si vous souhaitez modifier une journée, il suffit de la sélectionner :

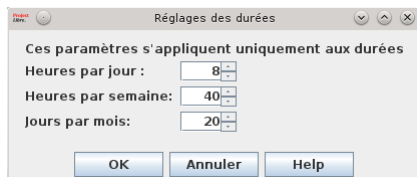
Pour modifier une période, il suffit de sélectionner « Temps de travail spécial », ensuite nous pourrions modifier les plages horaires, et le jour deviendra rouge sur le calendrier :



Maintenant, si vous souhaitez modifier les horaires sur une journée pour tout le projet, il vous suffit de sélectionner le jour concerné et faire la même opération que précédemment, ce qui donnera par la suite :



Si vous procédez ainsi, il ne faudra pas oublier de modifier les paramètres de durée, qui se trouvent avec le bouton « Options... » :



Vous avez aussi la possibilité de créer des types de calendriers, ce qui peut vous être utile si vous travaillez sur plusieurs projets avec des horaires identiques, et ainsi profiter des mises à jour.  
Pour cela, il suffit de cliquer sur « Nouveau... »

## 6 Exemple

Maintenant que nous allons passer à la construction du projet dans le logiciel, il faut que les différentes tâches soient connues.

Pour notre exemple, prenons le cas suivant :

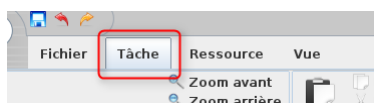
Description	Prédécesseurs	Durée (jour)
Libellé A		3
Libellé B	A	2
Libellé C		10
Libellé D	B, C	1
Libellé E	C	5
Libellé F	D, E	8
Libellé G	F	10



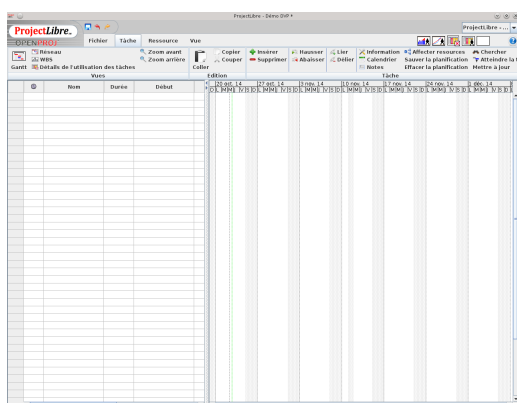
Et il est très important de bien définir toutes les tâches, cela n'en sera que plus facile pour utiliser et renseigner le programme par la suite.

## 7 Les tâches

Il suffit de sélectionner l'onglet « Tâche » :



Nous obtenons la fenêtre suivante :



### 7.1 Créer une tâche

Pour créer une tâche, il suffit de saisir les données dans la partie gauche de l'écran :

	Nom	Durée	Début	Fin	Coller	Edition
1	Libellé A	1 jour	22/10/14 08:00	22/10/14 09:00		

Ensuite, il suffit de continuer à saisir les données ou de cliquer sur « Information », ce qui ouvre la fenêtre suivante :


Sur cette fenêtre, il y a beaucoup d'informations qui sont réparties dans six onglets. Toutes ces informations ne sont pas à saisir au moment de la création, mais elles pourront être complétées ou modifiées au cours de la gestion et de l'organisation du projet.

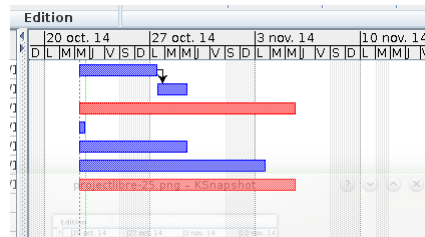
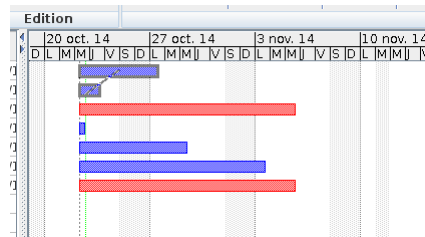
Dans notre cas et avec les éléments fournis dans le tableau précédent, je n'ai pour le moment saisi

que le nom des tâches et la durée, nous allons ainsi obtenir la liste suivante :

ID	Nom	Durée	Début	Fin
1	Libellé A	3 jours	20/10/14	23/10/14
2	Libellé B	2 jours	20/10/14	22/10/14
3	Libellé C	10 jours	20/10/14	30/10/14
4	Libellé D	1 jour	20/10/14	21/10/14
5	Libellé E	5 jours	20/10/14	25/10/14
6	Libellé F	8 jours	20/10/14	28/10/14
7	Libellé G	10 jours	20/10/14	30/10/14

Dans la partie gauche, nous avons les tâches, et dans la partie droite la représentation de la durée.

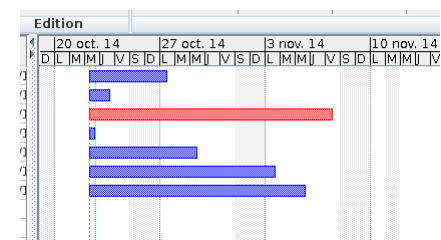
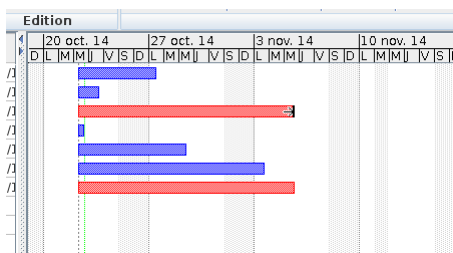
 La partie rouge représente le chemin critique, mais nous y reviendrons plus tard.



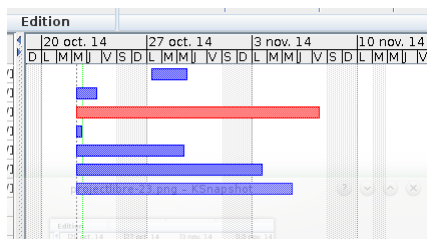
## 7.2 Gérer et organiser les tâches

Passons maintenant à la partie gestion et organisation des tâches. Nous savons que toutes les données sont modifiables avec la fenêtre des propriétés, mais certaines le sont sur le graphique :

- la durée peut être modifiée, il suffit pour cela de se placer sur le bord du segment et de faire glisser en moins ou en plus :

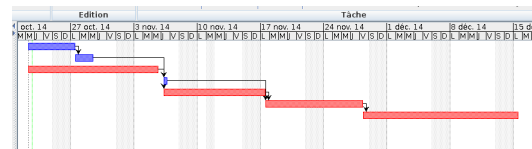


- les dates sont aussi modifiables sur le graphique, pour cela il nous suffit de sélectionner le segment et de le déplacer :



- la définition des précédents est aussi réalisable dans le graphique, il nous suffit de cliquer le segment de départ et de le faire glisser vers le segment d'arrivée :

La représentation finale de notre exemple donne donc :



Mettons-nous sur la tâche « Libellé F » et activons la fenêtre des propriétés dans l'onglet « Prédécesseurs » :

Prédécesseur	ID prédecesseur	Categorie	Décalage
Libellé D	4	FS	0 jours
Libellé E	5	FS	0 jours

## 7.3 La fenêtre information sur la tâche

### 7.3.a Onglet « Général »


#### Nom

Celui-ci peut-être modifié tout au long de la vie du projet. Il est préférable de bien expliquer à quoi correspond la tâche, cela n'en sera que plus facile pour la lecture globale du projet.

#### Dates et Durée

Ces trois éléments sont liés, il suffit d'en renseigner un pour que les deux autres se mettent à jour.

Si vous modifiez une des deux dates, c'est la durée qui évolue.

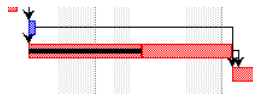
 La durée est exprimée en jour, mais si vous saisissez :

- 1 s deviendra 5 jours ;
- 1 M deviendra 20 jours.

« s » est l'abréviation de semaine et « M » de mois.

### Pourcentage terminé

C'est l'avancement de la tâche. Il suffit de mettre une valeur allant de 0 à 100 sur l'état d'avancement de la tâche ; une fois renseigné, cela sera visible sur le graphique :



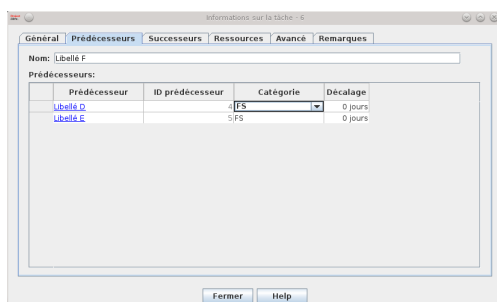
### Priorité

Cela permet de définir une priorité sur une tâche.

### Travail

Contient la durée de la tâche, calculée en fonction des plages horaires et des jours libres.

### 7.3.b Onglet « Prédécesseurs »



Sur cet onglet, nous allons pouvoir identifier les tâches qui devront être réalisées pour exécuter la tâche sélectionnée.

#### Prédécesseur

Indique le nom de la tâche qui précédera. Cela mettra à jour automatiquement les dates de début et de fin.

#### Catégorie

Normalement, une tâche ne peut commencer tant que l'autre n'est pas finie, mais il existe d'autres relations.

FS : c'est la plus couramment utilisée. Une tâche ne peut commencer si la précédente n'est pas terminée.

FF : dans ce cas-là, les deux tâches se termineront en même temps.

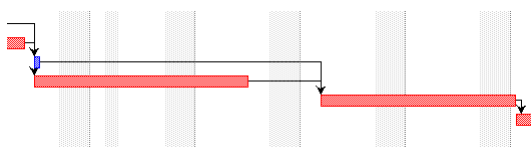
SS : dans ce cas-là, les deux tâches commenceront en même temps.

SF : dans ce cas-là, la tâche ne pourra commencer qu'au moment où l'autre sera terminée.

#### Décalage

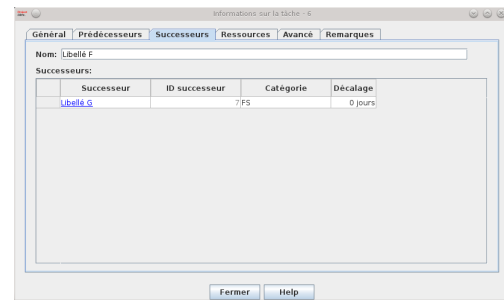
Cette valeur permet d'ajouter une durée à la tâche sans que celle-ci ne soit comprise dans son déroulement.

Par exemple, sur la tâche F, si nous mettons 2 sur la tâche E, voici ce que devient le graphique :



### 7.3.c Onglet « Successeurs »

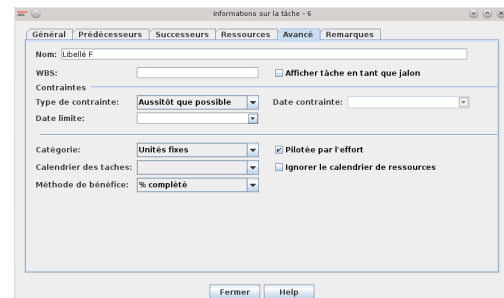
Il est identique à l'onglet « Prédécesseurs », mais pour les tâches qui devront s'exécuter une fois celle-ci finie :



### 7.3.d Onglet « Ressources »

Cet onglet sera abordé dans le paragraphe des ressources.

### 7.3.e Onglet « Avancés »



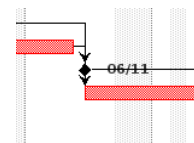
Sur cet onglet, nous allons pouvoir apporter des informations complémentaires sur la tâche.

#### WBS

C'est le Work Breakdown Structure (structure de découpage du projet).

#### Afficher tâche en tant que jalon

Il permet un événement sur une journée, il est représenté ainsi dans le graphique :



Très utile pour symboliser une réunion !

#### Contraintes

C'est une information complémentaire sur la date minimum de début d'une tâche. L'indicateur suivant apparaît alors :



Les autres informations étant des éléments permettant de calculer les charges.



### 7.3.f Onglet « Remarques »

Cette zone permet de saisir du texte, ce qui peut être pratique pour faire le suivi du projet. L'indicateur suivant apparaît alors :

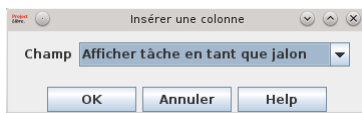


### 7.3.g Colonnes personnalisées

Dans la partie gauche de l'écran, des informations par défaut apparaissent. Nous avons la possibilité de les modifier en en ajoutant ou en en supprimant, il suffit pour cela de faire un clic droit sur un titre de colonne :



Dans le cas d'une insertion, la fenêtre suivante apparaît :



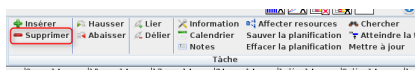
Il ne vous reste plus qu'à sélectionner la colonne souhaitée pour la voir apparaître.

## 7.4 Suppression / Insertion

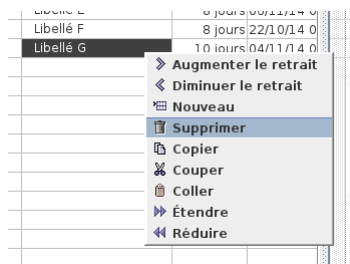
### 7.4.a Supprimer une tâche


Pour supprimer une tâche, il faut la sélectionner. Ensuite :

- dans la barre d'outils :



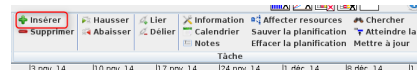
- dans la fenêtre des tâches avec un clic droit :



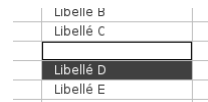
 Aucun message de validation n'apparaît au moment de la suppression.

### 7.4.b Insérer une tâche

Pour insérer une tâche, il faut se positionner sur la tâche suivante puis cliquer sur le bouton « Insertion » :



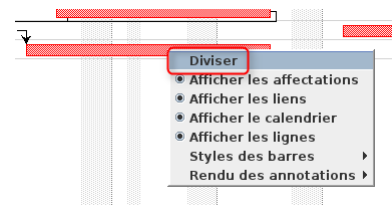
Ce qui donne :



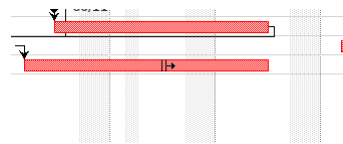
Il ne reste ensuite plus qu'à remplir les éléments comme pour la création d'une tâche.

### 7.4.c Fractionner une tâche

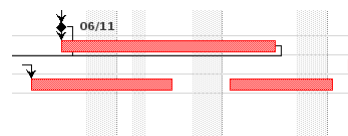
Nous avons aussi la possibilité de fractionner une tâche. Il nous suffit de faire un clic droit sur la tâche :



Ce qui nous donne :



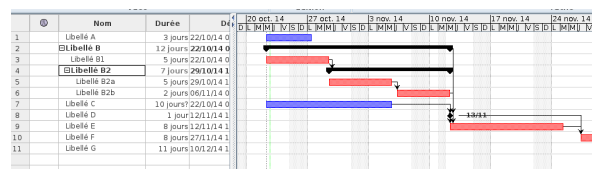
Ce qui donne finalement :



## 7.5 Arbre hiérarchique ou sous-tâche

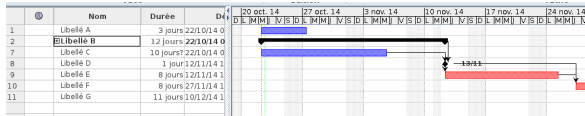
Si vous avez beaucoup de tâches à mettre dans votre projet, cela devient rapidement illisible, vous avez donc la possibilité de créer des arbres hiérarchiques ou sous-tâches.

Voici un exemple d'arbre hiérarchique et de sa représentation graphique :

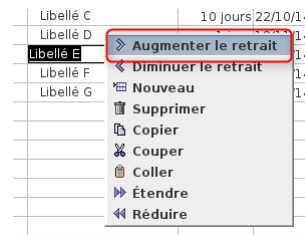


Dans la représentation graphique, les arbres ne sont plus représentés par des segments, mais par des traits. Ceux-ci tiennent compte des éléments qui sont inclus dans les niveaux inférieurs.

Cette utilisation permet de simplifier les affichages :



— par un clic droit :

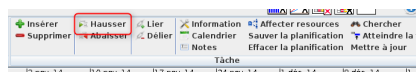


L'arbre sera automatiquement créé :



Pour créer un arbre hiérarchique, il faut se mettre sur la tâche et faire un clic droit, puis sélectionner la commande :

— dans la barre d'outils :



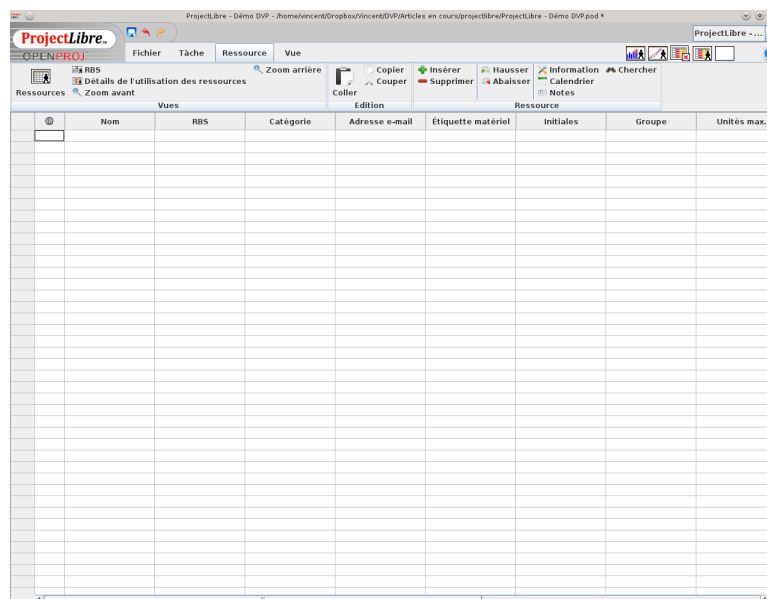
Au même niveau vous avez la commande qui fait l'inverse.

## 8 Les ressources

Il suffit de sélectionner l'onglet « Ressource » :



Nous obtenons la fenêtre suivante :

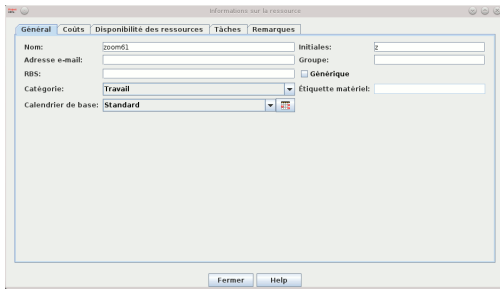


### 8.1 Créer les ressources

Pour créer une ressource, il suffit de saisir les données sur l'écran :

Ensuite, il suffit de continuer à saisir les données ou de cliquer sur « Information », ce qui ouvre la fenêtre suivante :

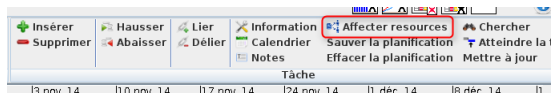




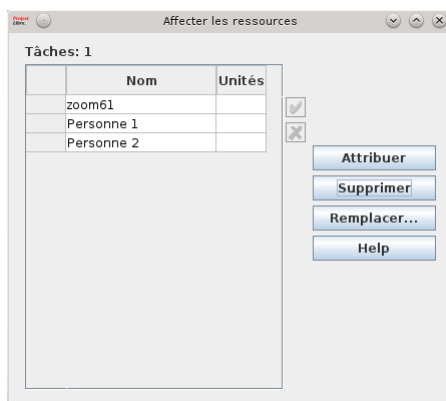
Sur cette fenêtre, il y a beaucoup d'informations qui sont réparties dans cinq onglets. Toutes ces informations ne sont pas à saisir au moment de la création, mais elles pourront être complétées ou modifiées au cours de la gestion et de l'organisation du projet.

## 8.2 Affecter les ressources

L'affectation des ressources dans la fenêtre des « Tâches » :




Il suffit de positionner sur une tâche et la fenêtre suivante qui apparaît est :

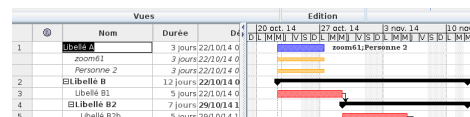


Ensuite, il suffit de sélectionner une ressource, de lui mettre ou pas une valeur à « Unités » et ensuite de cliquer sur « Attribuer », ce qui donne par la suite :



 Pour supprimer, il suffit de sélectionner la ressource et de cliquer sur « Supprimer ».

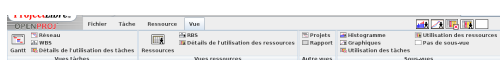
Et la représentation graphique devient donc :



## 9 Suivi du projet

ProjectLibre est un outil permettant de faire le suivi de vos projets. Il dispose pour cela de différents outils pour vous aider dans l'analyse et la diffusion des documents.

Ces différents états sont rassemblés dans l'onglet « Vue » :



### 9.1 Chemin critique

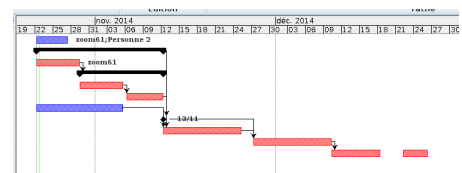
Le chemin critique désigne la liste ordonnée des opérations à réaliser pour obtenir le résultat demandé, sa durée donne la durée du projet.

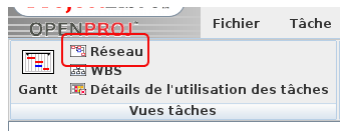
Il est visible tout au long du projet et il est identifié par la couleur rouge des segments :

### 9.2 PERT

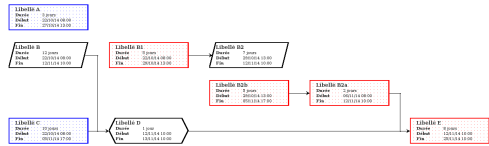
Ce terme signifie « Project Evaluation and Review Technique ». Cette méthode conventionnelle est utilisée en gestion de projets. Elle permet de représenter et d'analyser de manière logique les tâches du projet.

La commande se trouve dans la barre d'outils :





Ce qui nous donne :



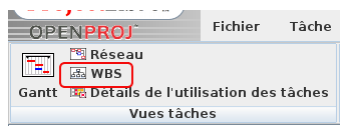
Chaque couleur a sa signification :

<p><b>Libellé B</b> Durée : 12 jours Début : 22/10/14 08:00 Fin : 12/11/14 10:00</p>	La bordure noire en parallélogramme permet d'identifier une tâche qui est composée de sous-tâches.
<p><b>Libellé B1</b> Durée : 5 jours Début : 22/10/14 08:00 Fin : 29/10/14 13:00</p>	La bordure rouge représente une tâche critique.
<p><b>Libellé A</b> Durée : 3 jours Début : 22/10/14 08:00 Fin : 27/10/14 13:00</p>	La bordure bleue correspondant à une tâche normale.
<p><b>Libellé D</b> Durée : 1 jour Début : 12/11/14 10:00 Fin : 13/11/14 10:00</p>	La bordure hexagonale correspond à un point bilan.

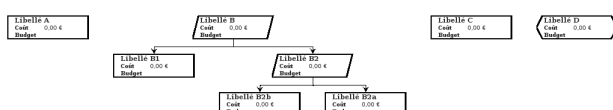
### 9.3 WBS

Work Breakdown Structure (WBS) en anglais se traduit par « structure de découpage du projet ». La WBS est constituée de différents éléments. Chacun correspond à une tâche ou à un ensemble de tâches du projet. Le premier élément d'une WBS est le projet lui-même et donc, il reçoit le nom du projet. Il a pour but d'aider à organiser le projet, à établir la planification de référence et le budget prévisionnel.

La commande se trouve dans la barre d'outils :



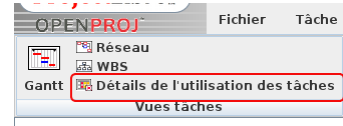
Ce qui nous donne :



### 9.4 Détails de l'utilisation des tâches

Sur cet état les ressources apparaissent sur chacune des tâches qui leur sont associées et pour chacune, nous avons la charge en heures.

La commande se trouve dans la barre d'outils :



Ce qui nous donne :

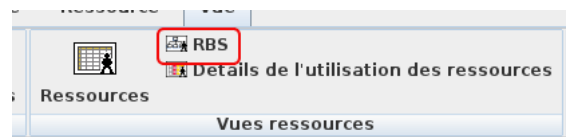
Item	Travail	Durée	Début	Fin	10	11	12	13	14	15	16	17	18	19	20
1	Libellé A	35 heures	22/10/2014 08:00	27/10/2014 13:00											
2	Libellé B	12 heures	22/10/2014 08:00	27/10/2014 13:00											
3	Libellé B1	5 heures	22/10/2014 08:00	29/10/2014 13:00											
4	Libellé B2	7 heures	22/10/2014 08:00	27/10/2014 13:00											
5	Libellé B3	40 heures	29/10/2014 08:00	30/10/2014 22:00											
6	Libellé B4	40 heures	29/10/2014 08:00	30/10/2014 22:00											
7	Libellé B5	40 heures	29/10/2014 08:00	30/10/2014 22:00											

Sur cet état, vous pouvez modifier les heures qui sont affectées à une ressource.

### 9.5 RBS

Ressource Breakdown Structure (RBS) en anglais se traduit par « Diagramme hiérarchique des ressources ». Elle permet de fournir les éléments financiers relatifs aux ressources.

La commande se trouve dans la barre d'outils :



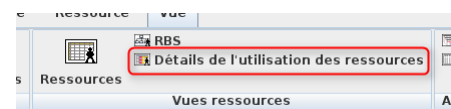
Ce qui nous donne :

Item	Cap	10/14	11/14	12/14	13/14	14/14	15/14	16/14	17/14	18/14	19/14	20/14
1	0.00 €											
2	0.00 €											
3	0.00 €											
4	0.00 €											
5	0.00 €											
6	0.00 €											
7	0.00 €											

### 9.6 Détails de l'utilisation des ressources

Sur cet état les ressources apparaissent et pour chacune des tâches qui lui correspondent, nous avons la charge en heures.

La commande se trouve dans la barre d'outils :



Ce qui nous donne :

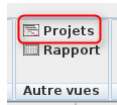
Item	Travail	Contour travailler	yes/no d.	10	11	12	13	14	15	16	17	18	19	20
1	000000	20 heures												
2	000000	20 heures/100	0 jours											
3	000000	40 heures/100	0 jours											
4	000000	7 heures												
5	000000	7 heures												
6	000000	7 heures												
7	000000	7 heures												

Sur cet état, vous pouvez modifier les heures qui sont affectées à une tâche de la ressource.

### 9.7 Projets

Cette commande permet de regrouper tous projets. Et cela contient la synthèse du projet en temps, coûts, etc.

La commande se trouve dans la barre d'outils :



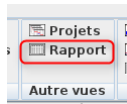
Ce qui nous donne :

Nom	Début	Fin	Responsable	Date de statut	Coût	Travail	Coût rest.	Travail r...
ProjectLibre - Démo DVP	2012/11/14 08:00	2012/11/14 10:00	jeanm1	2012/11/14 17:00	0,00 €	0,00 heures	0,00 €	0 heures

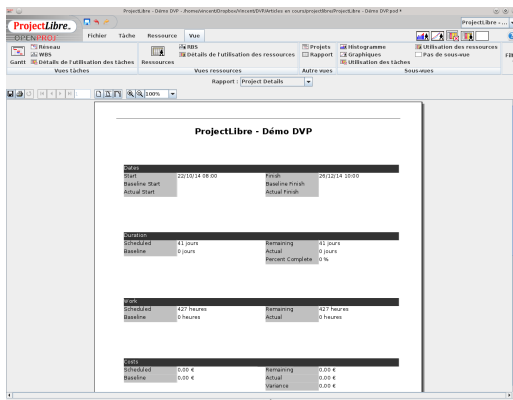
### 9.8 Rapport

Cette commande permet de regrouper tous les éléments du projet en cours.

La commande se trouve dans la barre d'outils :



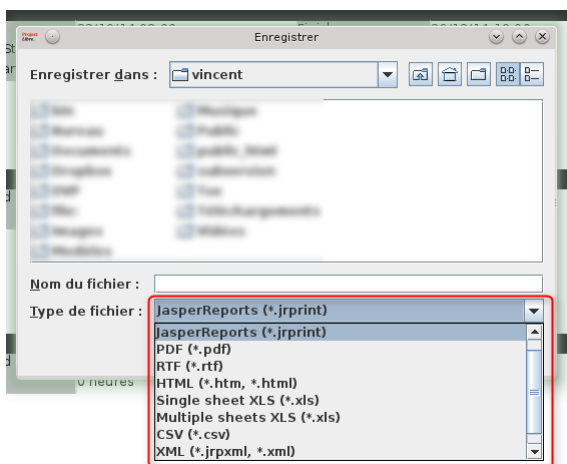
Ce qui nous donne :



Sur cet état, vous avez la possibilité d'enregistrer le document sous différents formats :

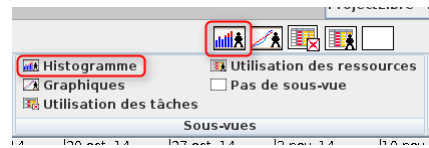


La liste des formats possibles est la suivante :

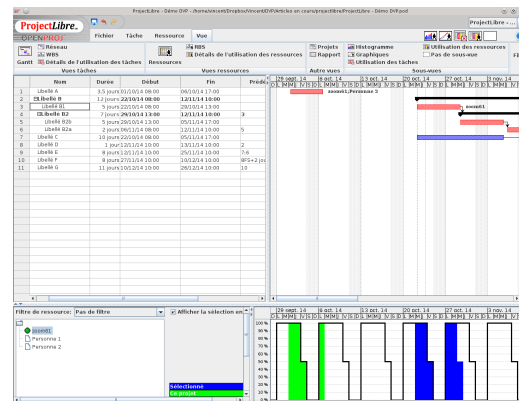


### 9.9 Histogramme

Cette commande se trouve ici :



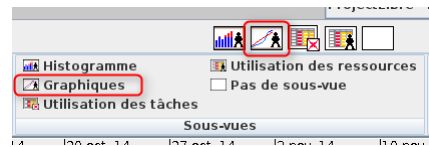
La fenêtre devient donc :



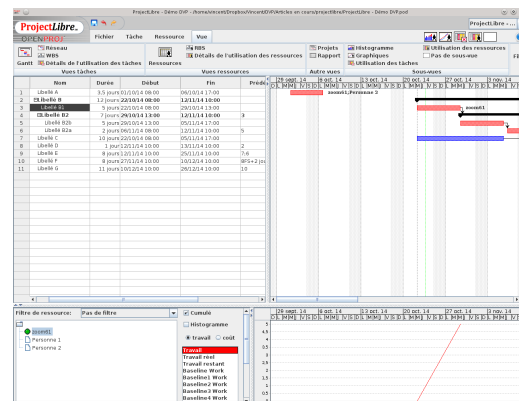
Le graphique évolue en fonction de la sélection de la ressource, de la tâche et de la position dans le calendrier.

### 9.10 Graphiques

Cette commande se trouve ici :



La fenêtre devient donc :



Le graphique évolue en fonction de la sélection de la ressource, de la tâche et de la position dans le calendrier.

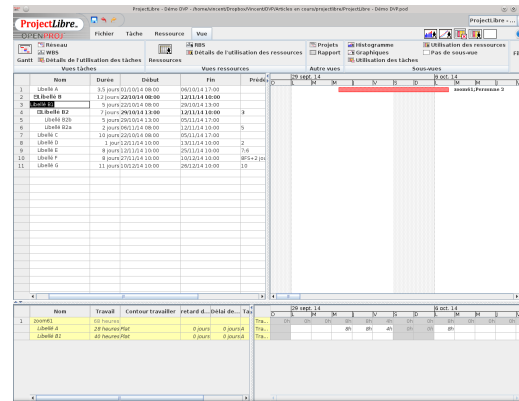
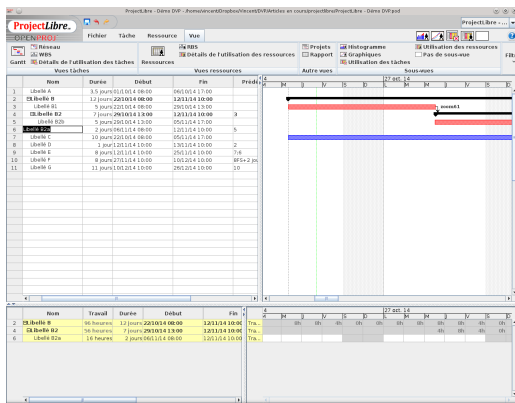


### 9.11 Utilisation des tâches

Cette commande se trouve ici :



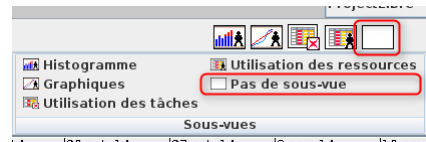
La fenêtre devient donc :



Le tableau évolue en fonction de la sélection de la ressource, de la tâche et de la position dans le calendrier.

### 9.13 Pas de sous-vue

Cette commande se trouve ici :

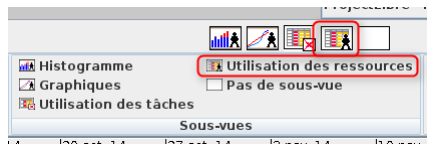


Cette commande permet d'effacer de l'écran les autres éléments correspondant aux commandes de la barre d'outils « Sous-vues »

Le tableau évolue en fonction de la sélection de la ressource, de la tâche et de la position dans le calendrier.

### 9.12 Utilisation des ressources

Cette commande se trouve ici :

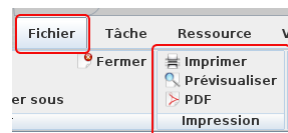


La fenêtre devient donc :

Retrouvez l'article de **Vincent Viale** en ligne : [lien 62](#)

### 9.14 Impression

Avec ProjectLibre, vous avez la possibilité d'imprimer les différents états ou de les convertir en PDF. La commande se trouve dans l'onglet « Fichier » :



Cette commande ne fonctionne que pour les états de détails et les éléments présents dans la barre d'outils « Sous-vues ».



# Perl

## Les derniers tutoriels et articles

# Encodage de caractères avec Perl

Cet article décrit les différents encodages de caractères, comment ils peuvent conduire à des problèmes et comment les gérer dans un programme Perl.

## 1 Introduction

Cela arrive trop souvent : un programme fonctionne correctement avec les caractères latins, mais il produit des caractères étranges ou illisibles dès qu'il faut utiliser des caractères chinois ou japonais, ou

des caractères latins modifiés tels que le « Umlaut » allemand Ä, Ö... ou les caractères scandinaves à et Ø.

## 2 ASCII

Pour comprendre le cœur du problème, il faut comprendre comment les caractères latins « normaux » et les autres caractères (ceux causant les problèmes) sont stockés.

Tout a commencé en 1963, avec le code ASCII, the « American Standard for Information Interchange ».

Puisqu'un octet contient 8 bits, le bit de poids fort d'un caractère ASCII est toujours à 0.

Le standard définit les caractères latins a à z en majuscules et en minuscules, les chiffres arabes 0 à 9, un espace appelé « blank », le retour chariot, et quelques caractères de contrôle et caractères spéciaux tels que %, \$. Les caractères non essentiels à l'usage quotidien des citoyens américains ne sont pas définis dans le codage ASCII, comme les lettres cyriliques, les caractères latins « décorés », les caractères grecs, et ainsi de suite.

## 3 Les autres encodages de caractères

Quand les gens ont commencé à utiliser des ordinateurs ailleurs qu'aux États-Unis, d'autres caractères ont été nécessaires. Dans les pays européens, l'ASCII fut réutilisé, et les 128 nombres inutilisés (le huitième bit de l'octet) ont été utilisés pour les caractères localement nécessaires.

En Europe de l'Ouest, l'encodage de caractères fut appelé « Latin-1 », puis plus tard standardisé sous le format ISO-8859-1. Latin-2 fut utilisé en Europe centrale, et ainsi de suite.

Dans chaque jeu de caractères Latin-\*, les premiers 128 caractères sont identiques à l'ASCII, donc,

cela peut être vu comme une extension de l'ASCII. Les autres 128 valeurs sont chacune mappées sur les caractères nécessaires à chaque région, chacun dans leur jeu de caractères utilisés.

Dans d'autres endroits du monde, d'autres encodages de caractères ont été développés, comme l'EUC-CN en Chine et le Shift-JIS au Japon.

Ces jeux de caractères locaux sont vraiment limités. Quand l'euro fut introduit en 2001, beaucoup de pays européens avaient un symbole pour leur monnaie qui ne pouvait pas être exprimé avec l'encodage de caractères traditionnels.

## 4 Unicode

Les jeux de caractères mentionnés ci-dessus ne peuvent encoder qu'une partie de tous les caractères possibles, ce qui rend quasi impossible la création de documents contenant des lettres de différentes écritures.

tures.

Dans une tentative d'unifier toutes les écritures dans une seule norme, le consortium Unicode fut créé, et il commença à collecter tous les caractères

connus, et à leur assigner chacun un nombre unique, appelé « point de code ».

Le « point de code » est habituellement écrit comme un nombre hexadécimal de quatre ou six chiffres, comme U+0041. Le nom correspondant à ce caractère est : « LATIN SMALL LETTER A ».

En dehors des lettres et autres « caractères de base », il y a aussi des accents aigus ou graves, des décorations (exemple : " ou ) qui peuvent être ajoutés aux caractères de base.

## 5 Formats de transformation Unicode

Le concept des points de code Unicode et des graphèmes est complètement indépendant de l'encodage.

Il y a différentes façons d'encoder ces points de code, et leurs mappings de points de code en octets sont appelés « Unicode Transformation Formats ». Le plus connu est l'UTF-8, qui est un format basé sur des octets qui utilisent les valeurs de 0 à 255. Dans le monde de Perl, il y a aussi une version nommée UTF8 (sans le trait d'union). Le module Perl *Encode* distingue les deux versions.

Si le caractère de base est suivi par un ou plusieurs de ces caractères de marquage, ce composé forme un caractère logique nommé « graphème ».

Il est à noter que beaucoup de graphèmes pré-composés existent pour les caractères qui sont définis dans d'autres jeux de caractères, et ceux-ci sont souvent mieux supportés par les logiciels courants que les caractères de base combinés aux caractères de marquage.

Windows utilise principalement l'UTF-16, qui utilise au moins 2 octets par point de code, pour les plus grands points de code, il utilise 4 octets. Il y a plusieurs variantes d'UTF-16, qui sont marquées par LE pour Little-Endian et BE pour Big-Endian (voir Endianness : lien 63).

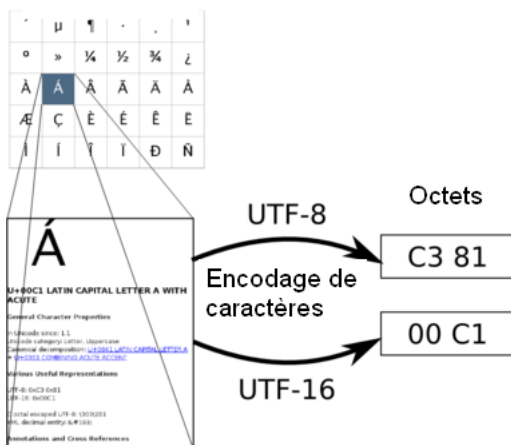
UTF-32 encode chaque point de code avec 4 octets. C'est le seul encodage à longueur fixe pouvant implémenter entièrement l'étendue des caractères Unicode.

points de code	Caractères	ASCII	UTF-8	Latin-1	ISO-8859-15	UTF-16
U+0041	A	0x41	0x41	0x41	0x41	0x00 0x41
U+00c4	Ä	-	0xc3 0x84	0xc4	0xc4	0x00 0xc4
U+20AC	€	-	0xe3 0x82 0xac	-	0xa4	0x20 0xac
U+c218	ᵀ	-	0xec 0x88 0x98	-	-	0xc2 0x18

La lettre dans la dernière ligne est la syllabe Han-geul SU, et votre navigateur ne peut l'afficher correctement que si vous avez les polices asiatiques appropriées installées.

Unicode définit un répertoire de caractères et ses propriétés. Les encodages de caractères UTF-8 et UTF-16 définissent une façon d'écrire ceux-ci en séquence courte d'octets.

### Répertoire de caractères



## 6 Perl 5 et les encodages de caractères

Les chaînes Perl peuvent être utilisées pour contenir des chaînes de caractères ou des données binaires. Avec une chaîne donnée, vous n'avez aucun mécanisme permettant de déterminer si elle contient du texte ou des données binaires, vous devez en garder une trace vous-même.

L'interaction avec l'environnement (comme lire un fichier, des données sur STDIN ou les imprimer) traite les chaînes comme des données binaires. La même chose est vraie pour les valeurs de retour de beaucoup de fonctions internes (comme `gethostbyname`) et les variables spéciales qui gardent les informations pour votre programme (`%ENV` et `@ARGV`).

Les autres fonctions internes qui interagissent avec du texte (comme les expressions régulières, `uc` et `lc`) traitent les chaînes comme du texte, ou plus exactement comme une liste de points de code.

Avec la fonction `decode` du module `Encode` ([lien 64](#)), vous pouvez décoder des chaînes binaires pour être sûr que les fonctions de traitement de texte fonctionnent correctement.

Toutes les opérations devraient fonctionner sur des chaînes qui ont été décodées par `Encode::decode` (ou avec les façons décrites ci-dessous). Autrement les fonctions de traitement de texte considèrent que les chaînes sont stockées en Latin-1 et donneront un résultat incorrect dans un autre encodage.

Notez que `cmp` compare uniquement les caractères non ASCII par des numéros de points de code, ce qui peut donner des résultats inattendus. En général, l'ordre est dépendant du langage utilisé et vous devez donc utiliser `use locale` pour trier des chaînes selon les règles naturelles du langage utilisé. Par exemple, en allemand, l'ordre de tri souhaité sera 'a' lt 'ä' and 'ä' lt 'b', tandis que la comparaison par numéro de points de code donnera 'ä' gt 'b'.

```

1  #!/usr/bin/perl
2  use warnings;
3  use strict;
4  use Encode qw(encode decode);
5
6  my $enc = 'utf-8';    # Ce programme est
7                      # encodé et enregistré en UTF-8
8  my $str = "A\n";
9
10 # Byte strings:
11 print lc $str;      # affiche 'A', lc
12                   # n'a aucun effet
13
14 # Chaîne de caractères
15 my $text_str = decode( $enc, $byte_str );
16 ;
17 $text_str = lc $text_str;
18 print encode( $enc, $text_str );    #
19                                     # affiche 'a', lc fonctionne

```

Il est fortement recommandé de convertir toutes les entrées en chaînes texte, de travailler avec ces chaînes texte et les mettre secrètement en chaînes d'octets à la sortie ou sur le support de stockage.

Sinon, vous pouvez très vite être induit en erreur et perdre les traces des chaînes qui sont des chaînes d'octets de celles qui sont des chaînes de texte.

Perl offre des couches d'entrée/sortie (IO Layers) qui ont des mécanismes de conversion automatique, soit globalement, soit par descripteur de fichier.

```

1  # IO layer: $handle décode désormais
2  # toutes les chaînes lors de la
3  # lecture
4  open my $handle, '<:encoding(UTF-8)',
5  $file;
6
7  # same
8  open my $handle, '<', $datei;
9  binmode $handle, ':encoding(UTF-8)';
10
11 # tout appel à open() utilise
12 # automatiquement :encoding(iso-8859-1
13 # )
14 use open ':encoding(iso-8859-1)';
15
16 # Toutes les chaînes de caractères du
17 # script sont interprétées comme des
18 # chaînes de texte:
19 use utf8;
20
21 # considère que le script doit être
22 # stocké en UTF-8
23
24 # Récupère les paramètres régionaux (les
25 # "locales") de l'environnement et
26 # demande à STDOUT de faire la
27 # conversion vers ces paramètres ré
28 # gionaux
29 use PerlIO::locale;
30 binmode STDOUT, ':locale';
31
32 # Toutes les entrées/sorties avec l'
33 # encodage courant
34 use open ':locale';

```

Une attention particulière doit être portée à : `utf8`, qui est souvent citée dans les exemples de code et de documentation : cette fonction garantit que l'entrée sera valide en UTF-8, et vous n'avez aucun moyen de savoir dans votre programme si c'est réellement le cas. Si ça ne l'est pas, c'est une source de trou de sécurité subtil, regardez cet article ([lien 65](#)) pour plus de détails. Ne l'utilisez jamais comme une couche d'entrée, utilisez `encoding(UTF-8)` à la place.

Le module et `pragma utf8` vous permet également d'utiliser des caractères non ASCII dans des noms de variables ou de modules. Mais attention, ne le faites pas pour les packages et les modules, cela ne fonctionne pas correctement. Considérez également que tout le monde n'a pas de clavier autorisant la saisie simple de caractères non ASCII, donc la maintenance de votre code s'en trouvera compliquée.

## 7 Tester votre environnement

Vous pouvez utiliser le petit script suivant dans votre terminal. Il est très centré sur l'Europe, mais vous pouvez l'adapter à l'encodage de caractères que vous utilisez où vous vivez.

```

1  #!/usr/bin/perl
2  use warnings;
3  use strict;
4  use Encode;
5
6  my @charsets = qw(utf-8 latin1 iso-8859-
   15 utf-16);
7
8  # quelques points de code non non-ASCII
9  my $test = '€: ' . chr(220) . ' '; Euro:
   ' . chr(8364) . "\n";
10
11 for (@charsets) {
12     print "$_: " . encode($_, $test);
13 }

```

Si vous lancez ce programme dans un terminal, une seule ligne sera affichée correctement, et sa première colonne correspond à l'encodage de votre terminal.

Le signe euro € n'est pas dans la table d'encodage Latin-1, donc si votre terminal est dans cet encodage, le signe euro n'apparaîtra pas correctement.

Les terminaux Windows utilisent principalement les encodages cp\*, par exemple cp850 ou cp858 (seulement disponibles dans les nouvelles versions de *Encode*) pour les installations allemandes. Le reste de l'environnement utilise les encodages Windows-\*, par exemple Windows-1252 pour plusieurs localisations de l'Europe de l'Ouest.

```
1 Encode->encodings(":all");
```

retourne une liste des encodages disponibles.

## 8 Dépannage

### 8.1 « Caractères larges à l'impression »

Quelquefois, vous pourriez voir ce message d'avertissement : `Wide character in print`.

Cela signifie que vous avez essayé d'utiliser des données d'une chaîne décodée dans un contexte où seules des données binaires ont un sens. Vous pouvez supprimer le *warning* en utilisant une couche de sortie appropriée, ou en encapsulant d'abord la chaîne incriminée avec `Encode : :encode`.

### 8.2 Inspecter des chaînes

Quelquefois, vous voulez vérifier si une chaîne de source inconnue a déjà été décodée. Comme Perl n'a pas de types de données séparés pour les chaînes binaires et les chaînes décodées, vous ne pouvez pas faire ceci de façon fiable.

Mais vous pouvez pallier cela en utilisant le module `Devel : :Peek` : [lien 66](#)

```

1  use Devel::Peek;
2  use Encode;
3  my $str = "a";
4  Dump $str;
5  $str = decode("utf-8", $str);
6  Dump $str;
7  Dump encode('latin1', $str);
8
9  __END__
10 SV = PV(0x814fb00) at 0x814f678
11 REFCNT = 1
12 FLAGS = (PADBUSY, PADMY, POK, pPOK)
13 PV = 0x81654f8 "\303\244"\0
14 CUR = 2
15 LEN = 4
16
17 SV = PV(0x814fb00) at 0x814f678
18 REFCNT = 1
19 FLAGS = (PADBUSY, PADMY, POK, pPOK, UTF8)
20 PV = 0x817fcf8 "\303\244"\0 [UTF8 "a"]

```

```

21 CUR = 2
22 LEN = 4
23
24 SV = PV(0x814fb00) at 0x81b7f94
25 REFCNT = 1
26 FLAGS = (TEMP, POK, pPOK)
27 PV = 0x8203868 "\344"\0
28 CUR = 1
29 LEN = 4

```

La chaîne UTF8 dans la ligne commençant par `FLAGS =` montre que la chaîne a déjà été décodée. La ligne commençant par `PV =` contient les octets et entre parenthèses les points de code.

Mais il y a un gros avertissement. Juste, car le flag `UTF8` n'est pas présent et qui ne veut pas dire que la chaîne texte n'a pas été décodée. Perl utilise soit `Latin1`, soit `UTF-8` en interne pour stocker les chaînes, et la présence de ce flag indique que chacun a été utilisé.

Cela implique aussi que votre programme n'est écrit qu'en Perl (et n'a pas de composant XS). Et c'est presque une erreur à coup sûr de compter sur la présence de ce flag : [lien 67](#). Vous ne devriez pas vous préoccuper de comment Perl stocke ses chaînes.

### 8.3 Modules bogués

Une source commune d'erreurs provient de modules bogués. Le `Pragma encoding` semble très tentant.

```

1 # conversion automatique grâce aux param
   ètres régionaux (locale)
2 use encoding ':locale';

```

Mais sous l'effet de `use encoding`, certaines fonctions `AUTOLOAD` s'arrêtent de fonctionner, et les modules ne sont pas *Thread safe*.

## 9 Encodage de caractères sur l'internet

Quand vous écrivez un script CGI, vous devez choisir un encodage de caractères, imprimer toutes vos données dans cet encodage, et l'écrire dans les en-têtes HTTP.

Pour la plupart des applications, l'UTF-8 est un bon choix, puisque vous pouvez arbitrairement coder les points de code Unicode avec. D'autre part, le texte anglais et la plupart des autres langages européens sont encodés vraiment efficacement.

L'HTTP offre l'en-tête `Accept-Charset-Header` dans lequel le client peut indiquer au serveur quel encodage il peut supporter. Mais si vous vous en tenez aux encodages communs comme l'UTF-8 ou le Latin-1, tous les user agents vont les comprendre, donc il n'est pas vraiment nécessaire de vérifier cet en-tête.

Les en-têtes HTTP eux-mêmes sont strictement en ASCII, donc toutes les informations envoyées dans l'en-tête HTTP (en incluant les URL et les cookies) doivent être encodées en ASCII si des caractères non ASCII sont utilisés.

Pour les fichiers HTML, l'en-tête ressemble typiquement à ceci :

```
1 Content-Type: text/html; charset=UTF-8.
```

Si vous envoyez un tel en-tête, vous n'avez qu'à échapper les caractères qui ont une signification particulière en HTML comme : `<`, `>`, `&` et, dans des attributs `"`.

## 10 Les modules

Il y a pléthore de modules Perl qui peuvent gérer le texte, et il y en a que quelques-uns de notables, et vous devez les rendre « Unicode-aware ».

### 10.1 LWP : :UserAgent et WWW : :Mechanize

Utilisez `$response->decode_content` à la place de seulement `$response->content`. Par ce biais, l'information d'encodage de caractères envoyée dans l'en-tête de réponse HTTP est utilisée pour décoder le corps de la réponse.

## 11 Thèmes avancés

Avec le charset de base et la connaissance de Perl, vous pouvez aller assez loin. Par exemple, vous pouvez faire une application web « Unicode safe », c'est-à-dire prendre soin que toutes les entrées possibles soient affichées correctement, dans n'importe quel

Un soin particulier doit être porté à la lecture des paramètres POST ou GET avec la fonction `param` dans le module CGI. Les vieilles versions (antérieures à 3.29) retournent toujours des chaînes d'octets. Les nouvelles versions retournent une chaîne texte si `charset("UTF-8")` a été appelé avant, et sinon une chaîne d'octets.

CGI.pm ne supporte pas d'encodage de caractères autre que l'UTF-8. Vous ne devez donc pas utiliser de routine de jeu de caractères et vous devez décoder explicitement les paramètres de chaînes vous-même.

Pour vous assurer que le contenu de formulaire du navigateur a été envoyé avec un charset connu, vous pouvez ajouter l'attribut `accept-charset` dans la balise `<FORM>`.

```
1 <form method="post" accept-charset="utf-8" action="/script.pl">
```

Si vous utilisez un système de templates, vous devez en choisir un qui sait comment gérer les encodages de caractères.

Les Template : :Alloy ([lien 68](#)), HTML : :Template : :Compiled (depuis la version 0.90 avec l'option `open_mode`) ([lien 69](#)), ou le Template Toolkit (avec l'option d'encodage dans le constructeur et un IO layer dans la méthode `process`) sont de bons exemples.

### 10.2 DBI

DBI laisse le traitement de l'encodage de caractères aux modules DBD : :(`driver`), ce que vous avez donc à faire dépend de la base de données que vous utilisez en arrière-plan. Pour la plupart de celles-ci, l'UTF-8 est l'encodage le mieux supporté.

- Pour **MySQL** et DBD : :mysql, passez l'option `mysql_enable_utf8=>1` à l'appel à `DBI->connect`.
- Pour **PostgreSQL** et DBD : :Pg, positionnez l'attribut `pg_enable_utf8` à 1.
- Pour **SQLite** et DBD : :SQLite, positionnez l'attribut `sqlite_unicode` à 1.

script que l'utilisateur est amené à utiliser.

Mais ce n'est pas tout ce qu'il faut savoir sur le sujet. Par exemple, le standard Unicode permet différents moyens de composer des caractères, vous devez donc les normaliser avant de comparer deux



chaînes. Vous pouvez lire plus à ce propos dans la FAQ Unicode normalization : [lien 70](#).

Pour implémenter un comportement spécifique à un pays dans des programmes, vous devez jeter un œil sur les locales du système. Par exemple, en Turquie, la minuscule de la lettre capitale I est

İ, U+0131 LATIN SMALL LETTER DOTLESS I, quand la majuscule de i est İ, U+0130 LATIN CAPITAL LETTER I WITH DOT ABOVE.

Une bonne lecture pour commencer à propos des locales est la `perldoc perllocale`.

## 12 Philosophie

Beaucoup de programmeurs confrontés à des problèmes d'encodage ont comme première réaction : « ça ne devrait pas marcher ? ». Oui, ça devrait

marcher, mais beaucoup trop de systèmes sont mal conçus au niveau de l'encodage et des jeux de caractères.

## 13 Mal conçu

« **mal conçu** » signifie la plupart du temps qu'un format de document, qu'une API ou un protocole autorise plusieurs encodages, sans une façon normalisée de transporter de conserver cet encodage avec l'information.

Un exemple classique est l'Internet Relay Chat (IRC), lequel spécifie qu'un caractère est un seul octet, mais pas quel encodage est utilisé. Cela marchait

bien du temps de l'encodage Latin-1, mais était voué à l'échec aussitôt que des personnes de différents continents allaient commencer à l'utiliser.

Actuellement, beaucoup de clients IRC essaient d'autodétecter l'encodage de caractères et le recodent selon la configuration de l'utilisateur. Cela fonctionne très bien dans certains cas, mais produit un résultat horrible là où cela ne fonctionne pas.

## 14 Autre exemple : le XML

L'Extensible Markup Language, communément connu par son abréviation XML, vous permet de spécifier l'encodage dans le fichier :

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

Il y a deux raisons pour lesquelles c'est suffisant :

1. L'information d'encodage est optionnelle. La spécification indique clairement que l'encodage doit être UTF-8 si l'information de celui-ci est absente, mais malheureusement beaucoup d'auteurs d'outils ne semblent pas le savoir et finalement émettent du Latin-1. (C'est bien sûr seulement en partie la faute de la spécification.);
2. Tout analyseur XML doit d'abord détecter au-

tomatiquement l'encodage pour pouvoir parser l'information encodée.

Le second point est vraiment important. Vous supposez que : « Il n'y a pas de problème, le préambule est juste de l'ASCII ».

Mais beaucoup d'encodages sont compatibles ASCII pour les 127 premiers octets (par exemple UTF-7, UCS-2 et UTF-16).

Ainsi, l'information d'encodage est disponible, le parser doit d'abord le deviner pour l'extraire correctement.

L'annexe aux spécifications XML contient un algorithme de détection ([lien 71](#)) qui peut gérer tous les cas classiques, mais ne comporte pas le support de l'UTF-7.

## 15 Comment faire les choses correctement : la signalisation Out-of-band

L'exemple XML au-dessus démontre que le support de l'information d'encodage est inclus dans le format de fichier lui-même, sauf si vous spécifiez un moyen de transporter l'information d'encodage au niveau de l'octet, indépendamment de l'encodage du reste du fichier.

Une solution possible aurait pu être de spécifier que la première ligne d'un fichier XML doit être encodée en ASCII, le reste du fichier étant encodé avec ce qui a été spécifié dans cette première ligne. Mais

c'est une solution horrible, un éditeur de texte normal pourrait afficher la première ligne de façon complètement fautive si le fichier est dans un codage incompatible avec l'ASCII. Bien sûr, c'est aussi incompatible avec les spécifications XML actuelles, et exigerait de nouvelles spécifications incompatibles qui casseraient toutes les applications existantes.

### Alors, comment bien faire les choses ?

La réponse est assez simple ; tout système fonctionnant avec des données texte doit stocker les mé-

adonnées séparément, ou tout stocker dans un encodage uniforme.

C'est tentant de tout stocker dans le même encodage. Et cela fonctionne très bien sur une machine locale, mais vous ne pouvez pas attendre de tout le monde d'être d'accord sur un seul encodage, donc tous les échanges de données doivent toujours incorporer l'information d'encodage. Et souvent, vous voulez stocker les fichiers originaux (par crainte de perte de données), donc vous devez garder l'information d'encodage quelque part.

Cette observation devrait avoir un énorme impact sur le monde de l'informatique : tous les systèmes de fichiers devraient permettre le stockage de l'information d'encodage comme métadonnée, et la retrouver facilement. Le même principe devrait s'appliquer pour les noms de fichiers, et les langages de programmation (au moins ceux qui veulent ne pas faire souffrir leurs utilisateurs) devraient transporter de façon transparente cette métainformation, et se préoccuper de tous les problèmes d'encodage.

Et là, ça pourrait fonctionner.

## 16 Pour en savoir plus

- W3c tutorial on character encodings in HTML and CSS : [lien 72](#) ;
- Perl Programming/Unicode UTF-8 wikibook : [lien 73](#) ;
- perlunitut, the Perl Unicode Tutorial : [lien 74](#).

## 17 Outils utiles

- gucharmap, la carte des caractères Unicode de Gnome : [lien 75](#) ;
- hexdump never lies (sur Debian se trouve dans le package bsdmainutils) : [lien 77](#)
- An UTF-8 dumper vous montre les noms des caractères non ASCII : [lien 76](#) ;
- iconv convertit des fichiers texte d'un encodage vers un autre : [lien 78](#)

Retrouvez l'article de **Moritz Lenz** traduit par **Christophe Louvet** en ligne : [lien 79](#)

# OpenOffice-LibreOffice



## Les derniers tutoriels et articles

# Créer un dépliant avec Texte (Writer)



Il existe des modèles pour réaliser des dépliants, mais je vais vous montrer au cours de ce tutoriel comment en créer un.

## 1 Introduction

Un dépliant, nommé aussi prospectus ou flyer, est une feuille de papier qui sert à faire passer un message publicitaire ou promotionnel. C'est le format idéal pour transmettre un message autour d'une promotion, et c'est surtout le moyen de communiquer un message fort.

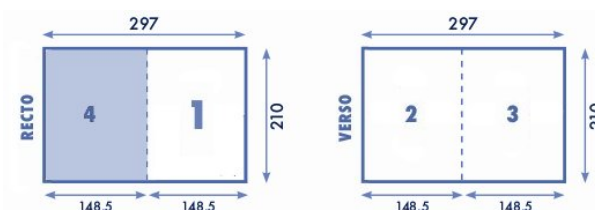
Pour cela, il faut pratiquer un pliage, et nous allons voir qu'il existe plusieurs formats de pliage, chacun avec ses particularités.

## 2 Les types de pliage

Le but d'un dépliant est d'utiliser toute la page, donc, les informations fournies sont données pour un format A4 (297 mm x 210 mm).

### 2.1 Le pli « simple »

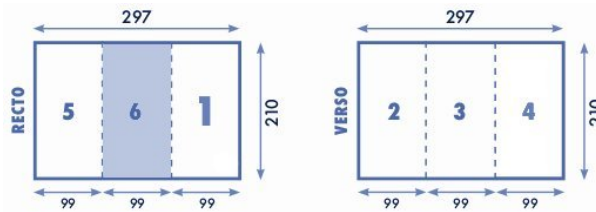
1 pli avec 2 volets



## 2.2 Le pli « accordéon »

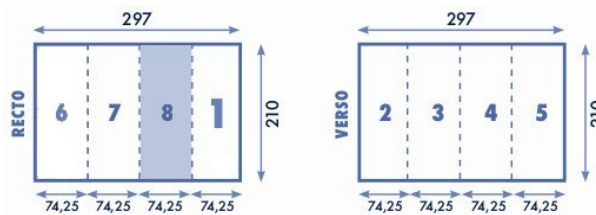
### 2.2.a Modèle 1

2 plis avec 3 volets



### 2.2.b Modèle 2

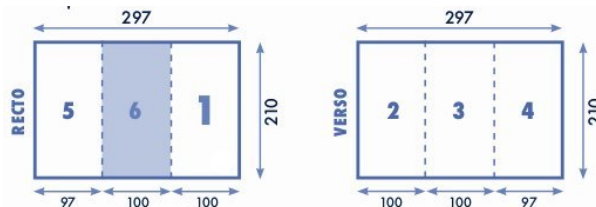
3 plis avec 4 volets



## 2.3 Le pli « roulé »

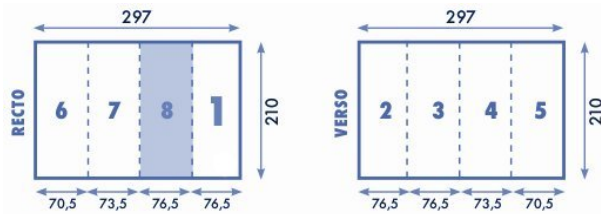
### 2.3.a Modèle 1

2 plis avec 3 volets



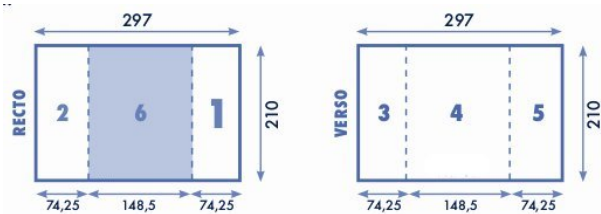
### 2.3.b Modèle 2

3 plis avec 4 volets



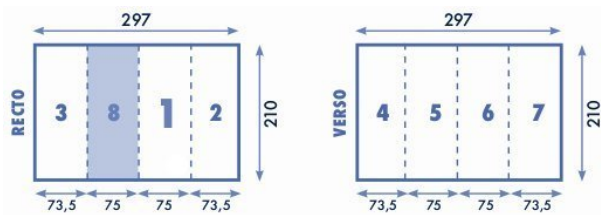
### 2.4 Le pli « fenêtre »

2 plis avec 3 volets



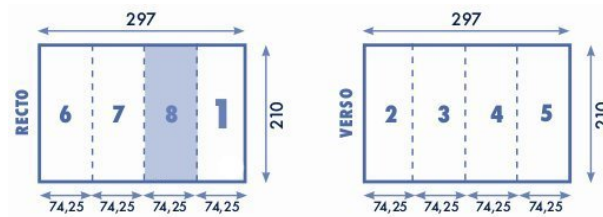
### 2.5 Le pli « portefeuille »

3 plis avec 4 volets



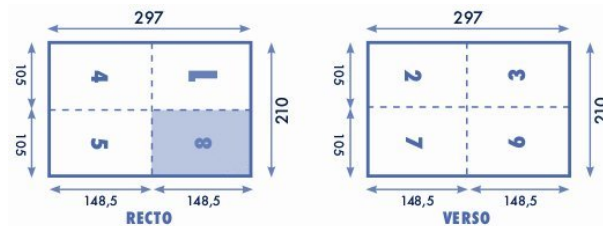
### 2.6 Le pli « économique »

3 plis et 4 volets



### 2.7 Le pli « croisé »

2 plis avec 4 volets



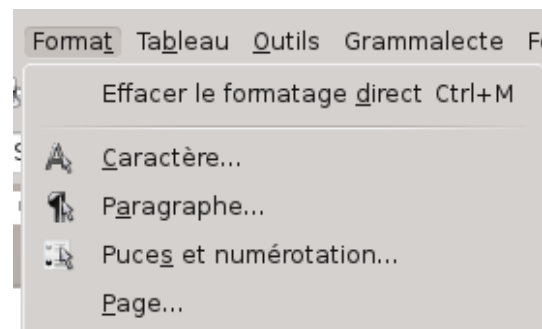
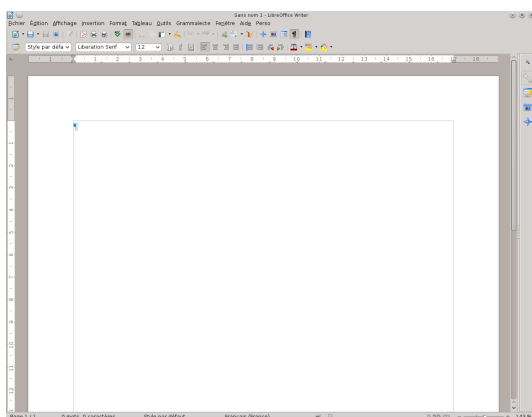
## 3 Avant de commencer

Une fois que vous avez défini la forme de votre dépliant et que vous savez quoi mettre dans les différentes parties, vous pouvez ouvrir « Texte » ou « Writer » sur un document vierge.

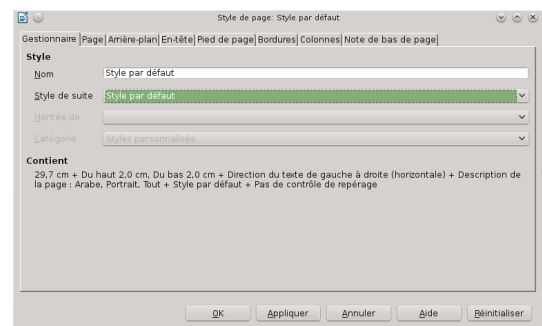
Pour faire notre dépliant, nous choisirons l'accordéon (modèle 1) :

## 4 Préparation du modèle

Une fois que vous avez ouvert un document vierge, vous obtenez ceci :



La fenêtre suivante apparaît :



Nous arrivons donc sur une page en format portrait.

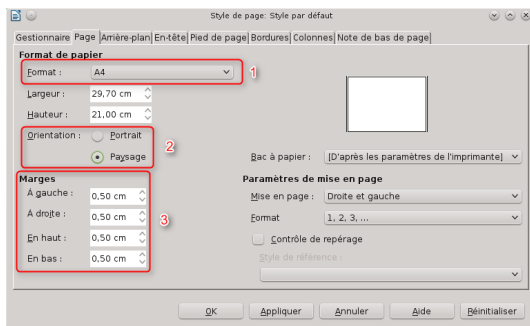
Maintenant, nous allons préparer la page afin de pouvoir y mettre les différents textes du dépliant.


Pour cela, nous devons aller dans « Format » et « Page... » :



## 4.1 La mise en page

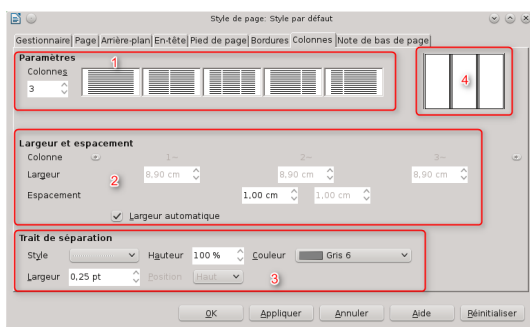
Vous devez sélectionner l'onglet « Page », dans cet onglet il faudra contrôler que la page est bien un format A4, lui indiquer l'orientation de la page (paysage pour notre cas), et définir les marges, la logique voudrait qu'il faille prendre 0, mais nous allons mettre 0,50 cm :



 Attention, si vous mettez « 0 » pour les marges, il faut savoir que les imprimantes ne sont pas capables de le faire. Le seul moyen sera alors de passer par un imprimeur pour obtenir le résultat attendu.

## 4.2 Définir le nombre de colonnes

Ensuite, les types de documents que nous avons choisis contiennent trois colonnes, cette donnée est modifiable dans l'onglet « Colonne » :



Cet onglet se décompose en quatre parties.

### 4.2.a Première partie

Cette partie permet de sélectionner le nombre de colonnes, soit en saisissant le nombre, soit en sélectionnant un modèle.

## 5 Préparation du document

Pour préparer le document final, il nous faut créer chaque partie du document.

La solution pour passer d'une colonne à une autre serait de faire autant de retours à la ligne que

### 4.2.b Deuxième partie

Cette partie permet de définir la largeur des colonnes, soit :

- en cochant « Largeur automatique » et les colonnes se dimensionnent en fonction de la largeur disponible, *que nous prendrons pour notre exemple* ;
- en saisissant la valeur colonne par colonne.

Dans cette partie, nous avons aussi la possibilité de mettre un espacement entre les colonnes. Comme nous avons mis un espace dans les marges, nous devons le reporter dans l'espacement, ce qui fait que nous allons mettre 1,00 cm.

### 4.2.c Troisième partie

Cette partie permet de rajouter un trait entre les colonnes, ce qui peut être intéressant dans le cas d'un pliage.

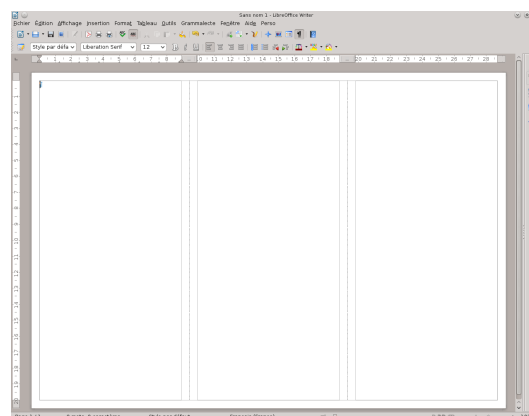
Dans notre cas, nous allons l'utiliser en prenant un style de trait, une épaisseur et une couleur claire. Ainsi, cela nous permettra un pliage plus facile et ne sera pas trop visible sur le document final.

### 4.2.d Quatrième partie

Cette partie nous donne la forme finale de notre document.

## 4.3 Le modèle

Une fois toutes les modifications effectuées, nous obtenons :



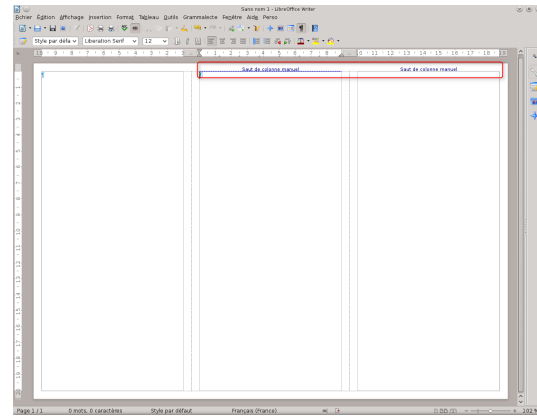
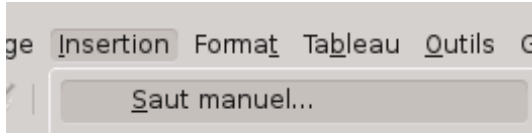
Nous y distinguons bien les trois colonnes, les espacements et les traits de séparation.

nécessaire pour y parvenir. Mais cette méthode est source de problèmes si l'on modifie le texte en rajoutant ou supprimant des mots.

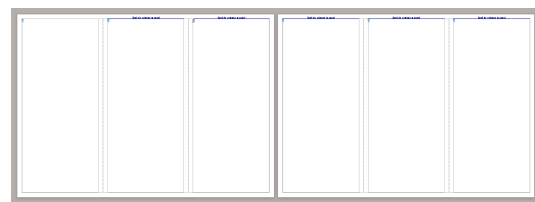
La solution consiste donc à utiliser les sauts de

colonnes que vous pouvez réaliser :

- avec l'appui des touches « Ctrl+Shift » avec la touche « Entrée » ;
- dans le menu « Insertion » et « Saut manuel... ».



Nous procédons de la même manière pour créer la seconde page, ce qui nous donne donc :

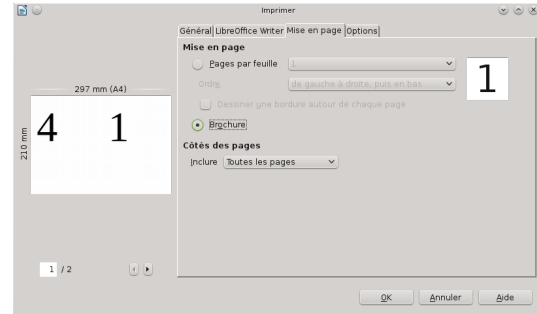


Finalement, sur la première page nous obtenons :

et il ne nous reste plus qu'à le compléter en se servant du modèle correspondant.

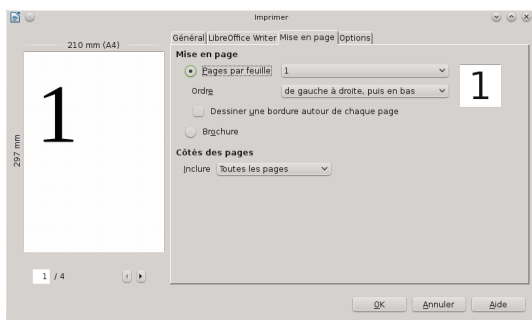
## 6 Variante

Pour le modèle pli « simple », vous pouvez créer votre modèle en mettant les pages les unes à la suite des autres, ce qui donne dans Texte :


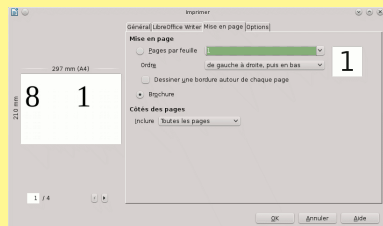


Ensuite, nous allons dans le menu « Fichier » et « Imprimer », et sélectionnons l'onglet « Mise en page » :

La page 4 vient se placer automatiquement devant la page 1.



Si vous souhaitez imprimer un document comme un livre, vous pouvez utiliser cette fonctionnalité, les pages seront alors mises en place automatiquement :

Dans cette fenêtre nous allons cocher la fonctionnalité « Brochure », ce qui nous donnera :

Retrouvez l'article de **Vincent Viale** en ligne : [lien 80](#)

# LibreOffice Calc requêter une base MySQL

On va voir dans ce tutoriel comment charger dans le tableur de LibreOffice des données provenant d'une base MySQL. Le PC client fonctionne sous une version Kubuntu et le serveur de BDD sur Debian (*mais ça n'a pas d'importance*).

## 1 Prérequis

- Un serveur MySQL installé.
- LibreOffice installé sur le poste client.

## 2 Permettre l'accès distant sur le serveur MySQL

Éditez le fichier `/etc/mysql/my.cnf`.

Afin de permettre les connexions depuis d'autres ordinateurs, modifiez la ligne `bind-address` par celle-ci :

```
1 bind-address = 0.0.0.0
```

```
1 #Redémarrez mysql
2 /etc/init.d/mysql restart
3 #puis connectez-vous à mysql
4 mysql -u root -p
```

Ajoutez un utilisateur ayant les droits depuis un poste distant (remplacez `localhost` par `%` dans la commande `GRANT`)

```
1 #pour un accès en lecture seulement :
```

```
2 GRANT SELECT ON votrebase.* TO 'user'@'%'
  IDENTIFIED BY 'votreMotDePasse';
3 #Si vous souhaitez un accès total à la
  base :
4 GRANT ALL ON votrebase.* TO 'user'@'%'
  IDENTIFIED BY 'votreMotDePasse';
```

Si nécessaire, ouvrez le pare-feu pour permettre les connexions MySQL (port 3306). Voici un exemple si votre Firewall est à base d'IPtables et que votre LAN est 192.168.0.\* :

```
1 iptables -A INPUT -p tcp --destination-
  port 3306 -m iprange --src-range 192
  .168.0.1-192.168.0.254 -j ACCEPT
```

## 3 Sur le poste client

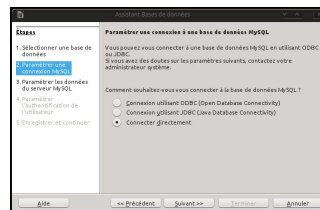
Installez le connecteur sur le client :

```
1 sudo apt-get install libreoffice-mysql-
  connector
```

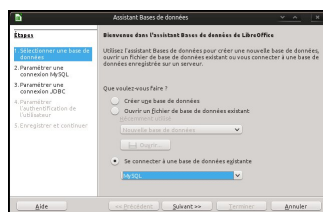
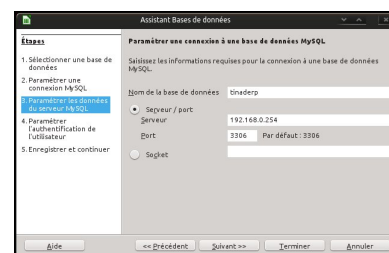
Voilà, on a tout ce qu'il faut. L'application LibreOffice Base servira d'intermédiaire entre Calc et le serveur MySQL. Depuis un Microsoft Office, MS Query aurait servi d'intermédiaire, mais il est plus transparent que oooBase.

Ouvrez LibreOffice -> Nouveau document-base de données.

Dans l'assistant de base de données, choisissez « Se connecter à une base de données existante » et « Mysql » comme ci-dessous :

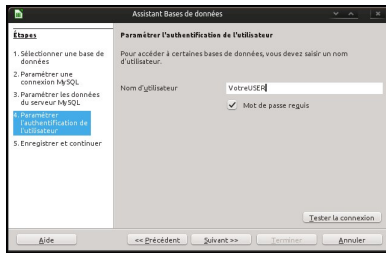


Ma base s'appelle tinaderp. Adaptez en fonction de la vôtre, ainsi que l'IP du serveur :



« Connecter directement » (c'est plus simple) :

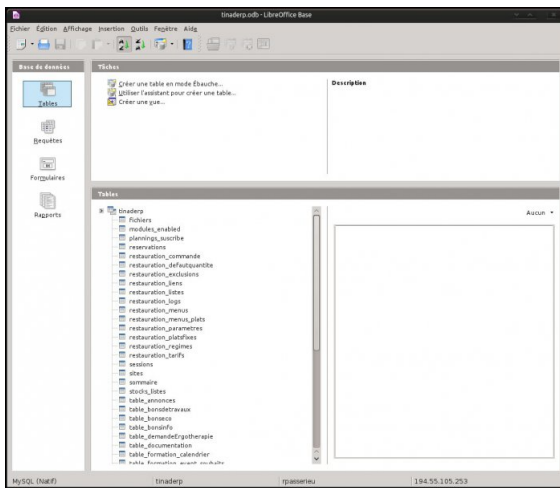
Puis l'utilisateur MySQL :



À l'étape suivante, surtout cochez « Oui, je souhaite que l'assistant référence la base de données ». Ce sera beaucoup plus facile pour retrouver les données dans le tableur. Puis cliquez sur terminer.

Il vous demande d'enregistrer le fichier. Je vous conseille de donner au fichier le nom de votre base de données.

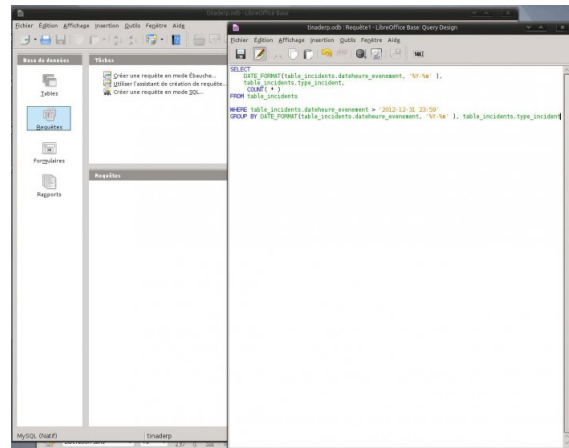
On arrive sur cette interface :



### 3.1 Créer quelques requêtes

Mon objectif ici est de pouvoir faire des statistiques. La BDD étant relativement brute. Si j'importe la base de données directement dans un tableur, ça va être pénible (*ceux qui ont fait déjà des formules de recherches matricielles dans un tableur me comprendront*). Je vais donc créer les requêtes SQL qui me feront les sous-totaux, les décomptages tout en liant les tables afin d'avoir les intitulés et pas les clés. Bref, je veux avoir les tableaux que je n'aurai qu'à sélectionner pour établir les graphiques. Si ce n'est pas votre cas, que le contenu de tables de façon brute vous suffit. Passez directement au chapitre suivant.

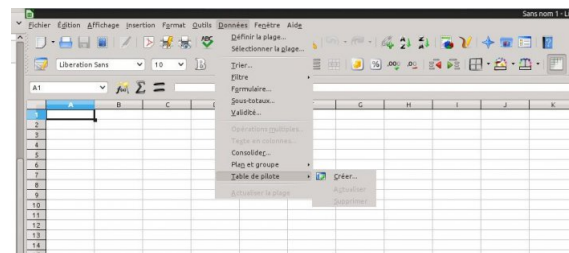
Requêtes -> Créer une requête en mode SQL :



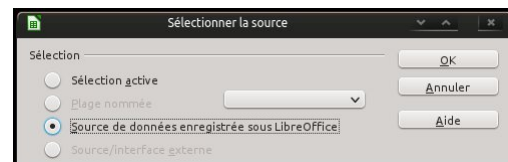
Testez la requête puis enregistrez-la. Donnez-lui un nom, le mien serait, par exemple : sommeParTypeParMois. Enregistrez votre fichier de base de données.

### 3.2 Dans le tableur Calc

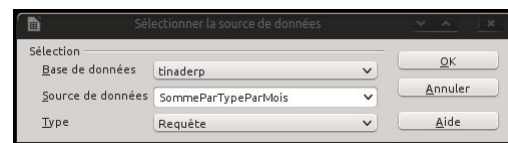
Données -> Table de pilote -> Créer



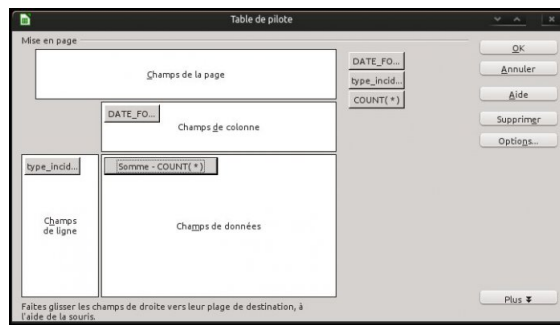
Source de données enregistrée sous LibreOffice :



- sélectionnez votre base (qui a le nom du fichier base) ;
- le type : feuille si vous voulez une table brute, ou requête si vous voulez récupérer les données d'une requête :



Organisez comme il faut les colonnes, les lignes et les données en faisant du glisser-déposer :



#### 4 Enjoy!

Somma - COUNT(*)	DATE_FO...	type_incid...	DATE_FO...	DATE_FO...	DATE_FO...	DATE_FO...	DATE_FO...	DATE_FO...	DATE_FO...	DATE_FO...	DATE_FO...	DATE_FO...	DATE_FO...	DATE_FO...	DATE_FO...
4	2013-01	2013-02	2013-03	2013-04	2013-05	2013-06	2013-07	2013-08	2013-09	2013-10	2013-11	2013-12	10		
68	71	54	78	68	67	69	104	85	76	110	48				
1		4	1	4	1					4	3				
	2	1	1	3	1		2	6	5	9	6	3			
15	17	16	13	16	13	16	17	13	26	13	18				
<b>Total Result</b>	<b>88</b>	<b>95</b>	<b>79</b>	<b>98</b>	<b>97</b>	<b>87</b>	<b>93</b>	<b>121</b>	<b>106</b>	<b>117</b>	<b>140</b>	<b>63</b>			

Retrouvez l'article de *Gnieark* en ligne : [lien 81](#)

# Java



## Les dernières news Java

# Oracle présente de nouvelles fonctionnalités de Java 9 et fixe les bases pour implémenter la modularité

Oracle a dévoilé de nouvelles fonctionnalités qui seront intégrées à Java 9, la prochaine version majeure de la plateforme de développement, qui sera disponible en 2016.

Java 9 introduira via la JEP 158 (Unified JVM Logging) ([lien 82](#)), un système d'enregistrement commun pour tous les composants de la JVM. Cela permettra une refonte complète de la façon dont Java signale les événements dans les sous-systèmes.

Plus de contrôle au niveau de la compilation sera offert par cette version (JEP 165 - More compiler controls : [lien 83](#)). Cette amélioration permettra aux développeurs de changer les options de compilation du compilateur JIT Hotspot afin de pouvoir effectuer des optimisations spécifiques.

Avec la JEP 214 (Remove GC Combinations Deprecated in JDK 8) ([lien 84](#)), trois fonctionnalités du garbage collection (ramasse-miettes) seront supprimées. Il s'agit de DefNew + CMS, ParNew + SerialOld et Incremental CMS. Ces fonctionnalités étaient déjà obsolètes dans Java 8.

Quelques petits changements seront apportés au projet Coin (JEP 213 : [lien 85](#)), qui avait été introduit dans Java 7. Pour rappel, le projet Coin apporte quelques changements linguistiques à Java.

En plus de ces fonctionnalités, Oracle a l'intention de finaliser avec le projet « Resolve Lint and Doclint Warnings » pour le nettoyage des avertissements, qui avait débuté il y a plusieurs années. Java 9 offrira également un support de Datagram Transport Layer Security, et des sorties HTML5 pour Javadoc. De nombreuses corrections seront également apportées pour améliorer la gestion des importations, et les classes dépréciées ne vont plus générer des alertes.

À ces spécifications, s'ajoutent d'autres JEP qui ont été présentées en août dernier, qui permettent de doter le langage de programmation d'une nouvelle « API lightweight JSON » pour la production et la consommation de documents JSON, de « HTTP2 Client » pour le support du HTTP 2.0 et des web

sockets et de « Process API Updates » qui permet d'améliorer le contrôle, la gestion et l'interaction avec les processus non Java.

Bien que le projet Jigsaw ne soit pas encore intégré au projet, Oracle a réaffirmé son engagement d'offrir la modularité dans Java 9. Les ingénieurs d'Oracle ont fixé les bases pour une mise en œuvre structurée du projet Jigsaw à travers la JEP 152, JEP 201 ([lien 86](#)) et JEP 220 ([lien 87](#)). Jigsaw représente une nouveauté très attendue. Mais, son développement fait face à plusieurs défis qui doivent être relevés avant son intégration.

Jigsaw apportera des gros changements au JDK. Il permettra de découper la bibliothèque d'exécution de base de Java en différents modules. Ainsi, une machine virtuelle Java (JVM) pourra fonctionner sans le support de certains packages de base.

Mark Reinhold, architecte en chef du groupe de la plateforme Java chez Oracle, a fait savoir qu'avec Jigsaw, le format JAR n'avait plus sa place dans le système d'exécution de Java. « Le format JAR a fait son temps, c'est le moment de passer à autre chose », avait affirmé celui-ci.

Une annonce qui n'a pas manqué de créer une certaine frayeur chez les développeurs. Mais, Oracle rassure. Bien que le système d'exécution de Java reposera sur les modules et non les fichiers jar, la plateforme continuera à prendre en charge et à exécuter les applications empaquetées dans les fichiers Jar. Avec le temps, les développeurs devront, cependant, migrer vers les nouveaux formats modulaires.

Le passage à un système de module aura également un impact important sur les outils de développement et les Framework.

À titre de rappel, les JEP sont les nouveaux processus permettant le développement et le test de fonctionnalités relatives au langage Java ou à sa machine virtuelle, sans recourir au processus complet de spécification (JSR). Par la suite, toute JEP qui a été implémentée avec succès sera intégrée à Java sous la forme de mise à jour ou de nouvelle version.

Commentez la news de **Hinault Romaric** en ligne : [lien 88](#)



# JDK 1.9 n'existera pas ! Un nouveau format de numérotation est proposé, le schéma actuel est trop contraignant

JDK 1.9, la prochaine version de kit de développement pour Java 9 n'existera pas. En effet, une proposition d'un nouveau schéma pour le numérotage des versions de Java est en cours d'étude.

Le nouveau schéma, proposé le 20 octobre dernier, sous la référence JEP 223 ([lien 89](#)), aura pour but de revoir le format de la chaîne de version du JDK de telle sorte qu'il soit plus facile de distinguer les révisions majeures, mineures, et les mises à jour de sécurité. Inspiré du principe du « versioning sémantique », il aura l'avantage d'être facilement compréhensible par les humains, et en même temps facilement analysable par les programmes.

Ce schéma aura le format suivant « \$Major.\$Minor.\$Security » où \$Major sera incrémenté lors d'un changement majeur des spécifications de la plateforme (comme l'arrivée de Java 8 par exemple). \$Minor sera incrémenté lors de changements mineurs de la version actuelle (comme la correction de bugs ou quelques révisions dans les API), et finalement, \$Security sera incrémenté pour chaque nouvelle mise à jour de sécurité. Ce dernier sera indépendant de \$Minor et ne sera remis à zéro que lorsque

\$Major changera.

Pour comprendre les avantages d'un tel format de numérotation, il faut regarder de plus près le schéma actuel : les numéros de versions avec des changements mineurs sont des multiples de 20. Les mises à jour de sécurité sont basées sur la version mineure précédente à laquelle on ajoute le nombre 5 (ou 6 si nécessaire afin de maintenir le numéro de mise à jour impair). Du coup : « JDK 7 Update 55 » et « JDK 7 Update 60 » contiennent toutes deux les mêmes mises à jour de sécurité, alors que n'importe quelle personne non prévenue croirait que « JDK 7 Update 60 » contient 5 mises à jour supplémentaires par rapport à « JDK 7 Update 55 ».

De plus, « JDK 7 Update 60 », « 1.7.0\_60 » et « JDK 7u60 » sont actuellement trois mêmes façons de représenter la même version, ce qui rend la comparaison entre les différentes versions fastidieuses pour les programmeurs, alors que le nouveau format proposé sera facilement analysé par cette expression régulière : « [1-9][0-9]\*(\.(0|[1-9][0-9]\*))\* ».

Du coup, ne vous étonnez pas si Java passe de « JDK 1.8 » directement à « JDK 9 » !

Commentez la news d'[Amine Horseman](#) en ligne : [lien 90](#)

## Le projet Jigsaw s'invite dans Java, les développeurs peuvent tester la modularité dans la dernière « Early Access » du JDK 9

Le projet Jigsaw fait, enfin, ses premiers pas dans le JDK. Les développeurs peuvent tester cette nouveauté dans une récente version de la « Early Access » du JDK 9.

L'objectif du projet Jigsaw est de concevoir et mettre en œuvre un système de modules standards pour Java SE, et d'appliquer ce système pour la plateforme elle-même et pour le JDK. Le projet Jigsaw permettra de découper la bibliothèque d'exécution de base de Java en différents modules. Cela devrait permettre à une machine virtuelle Java (JVM) de fonctionner sans le support de certains packages de base.

À la suite d'un travail préliminaire qui avait abouti à une proposition en 2008, le système de modules du projet Jigsaw allait être l'une des principales innovations de Java 7. Le projet avait été ensuite reporté à Java 8, puis finalement, Oracle avait annoncé sa sortie avec Java 9, prévue pour début 2016.

Oracle fait face à plusieurs défis dans le développement de ce projet qui permettra de redéfinir l'architecture de la plateforme. La principale difficulté à laquelle ont fait face les ingénieurs d'Oracle était liée aux dépendances qui existent entre les packages.

Cette préversion est basée sur Jigsaw m2, qui

implémente la JEP (JDK Enhancement Proposals) 200 (qui définit la structure modulaire du JDK), JEP 201 (qui réorganise le code source du JDK sous forme de module) et JEP 220 (qui restructure le runtime du JDK et JRE pour soutenir les modules).

Les développeurs en charge du projet ont débuté avec la JSR (Java Specification Requests) 376, qui permettra de rendre le système de module effectif dans la plateforme Java.

Il faut noter que le passage à un système de module aura également un impact important sur les outils de développement et les Framework. « Nous sommes conscients que ces changements vont briser certaines applications, en particulier les IDE et les outils de développement qui s'appuient sur la structure interne du JDK. », écrit Mark Reinhold, architecte en chef du groupe de la plateforme Java chez Oracle, dans un billet de blog. « Nous pensons cependant que les améliorations des performances, de la sécurité et de la maintenabilité donnent un sens à ces changements. »

Les ingénieurs d'Oracle ont d'ores et déjà contacté les développeurs des principaux environnements de développement pour s'assurer qu'ils ont

pris connaissance de ces changements et sont prêts, si nécessaire, à travailler avec ceux-ci.

Mark Reinhold avait déjà fait savoir qu'avec Jigsaw, le format JAR n'avait plus sa place dans le système d'exécution de Java. « Le format JAR a fait son temps, c'est le moment de passer à autre chose », avait affirmé celui-ci. La plateforme Java continuera, cependant, à prendre en charge et à exécuter les applications empaquetées dans les fichiers Jar.

En plus de l'intégration du projet Jigsaw, des modifications ont été également apportées à certaines JEP. Il s'agit notamment de la JEP 158 (Unified JVM Logging), qui introduit un système d'enregistrement commun pour toutes les composantes de la machine virtuelle Java. Cette JEP a été fortement influencée par ce qui existe dans la machine virtuelle JRockit d'Oracle.

La JEP 165 (Compiler Control) a aussi eu droit à des améliorations, tout comme la JEP 214 (Remove GC Combinations Deprecated in JDK 8) et la JEP 213 (Milling Project Coin).

La « Early Access » de JDK 9 est disponible en téléchargement sur le site du projet.

Commentez la news de **Hinault Romaric** en ligne : [lien 91](#)

## Les derniers tutoriels et articles

# Tutoriel sur l'API Cell de JavaFX - Comment customiser les contrôles utilisant l'API Cell tels que ComboBox, ListView, TableView, TreeView et TreeTableView.

Cet article vous montera les bases de l'utilisation de l'API Cell de JavaFX. Cette API constitue la base des contrôles dits « virtualisés » tels que *ComboBox*, *ListView*, *TableView*, *TreeView* et *TreeTableView*.

## 1 Introduction

Certains contrôles sont amenés à représenter à l'écran des ensembles d'objets, sous forme de liste, de boîte déroulante, de table ou encore sous la forme d'un arbre graphique.

Si on se contente des apparences par défaut, les contrôles existants sont, en l'état, tout à fait capables d'afficher de simples lignes de texte contenues dans des instances de *String*. Cependant, les choses se corsent lorsqu'on désire afficher des objets plus complexes dont la méthode *toString()* ne retourne pas forcément ce qu'on désire afficher à l'utilisateur. Ici, nous avons créé des instances d'une classe *Car*

que nous avons construite de toute pièce et nous les affichons dans une *ComboBox*. Comme nous n'avons pas redéfini la méthode *toString()* dans cette classe, l'affichage n'est... pas très sexy dirons-nous...

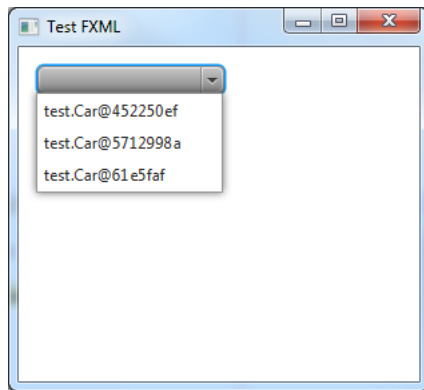


Figure 1 - Une JComboBox ayant un affichage très quelconque.

De plus, dans une application moderne, RIA

## 2 Rappel : Swing

Petit retour en arrière pour les aficionados de Swing. Swing utilise un concept assez proche de celui des cellules : un *renderer* ou encore tampon encreur. Un unique *cellRenderer* est créé et utilisé pour représenter chaque élément de la liste : à tour de rôle, il va endosser un élément et changer son apparence pour le représenter. Il s'agit de l'application du patron du poids-mouche : [lien 92](#). Cependant, l'implémentation dans Swing a un très gros défaut : elle n'est pas performante quand on l'utilise sur des conteneurs ayant de nombreux éléments. En effet le *renderer* est appelé sur l'intégralité des éléments !

Dans l'exemple suivant, nous plaçons 3000 éléments dans une *JList* (liste graphique), elle-même placée dans un *JScrollPane* (zone d'affichage restreint avec barre de défilement) :

```

1 final DefaultListModel model = new
  DefaultListModel();
2 for (int index = 0 ; index < 3000 ;
  index++) {
3   model.addElement(index);
4 }
5 final JList list = new JList();
6 list.setModel(model);
7 list.setCellRenderer(new
  DefaultListCellRenderer() {
8   private int called = 0;
9
10  @Override
11  public Component
    getListCellRendererComponent(JList
    <?> list, Object value, int index,
    boolean isSelected, boolean
    cellHasFocus) {
12    called ++;
13    System.out.println(called);
14    return super.
    getListCellRendererComponent(list
    , value, index, isSelected,
    cellHasFocus);
15  }
16 );
17 final JScrollPane listScroll = new
  JScrollPane(list);

```

(Rich Internet Application), ou encore mobile, on ne se contentera plus guère d'utiliser seulement du texte : ainsi, une liste de contacts dans une application de messagerie peut être amenée à afficher le portrait du dit contact ou encore changer la police ou la couleur du contact en fonction de son statut en ligne. De même, une table qui liste des produits dans un magasin peut afficher une image ou photo de ce produit, un hyperlien vers le site du fabricant ou encore une barre ou des étoiles montrant sa popularité auprès des clients.

C'est ici que l'API *Cell* (de l'anglais « cell » qui signifie « cellule ») entre en jeu : elle permet de rapidement customiser l'apparence d'une donnée dans un contrôle pour afficher la bonne valeur, le rendre plus attrayant ou encore le rendre interactif.

```

18 final JFrame frame = new JFrame();
19 frame.setDefaultCloseOperation(JFrame.
  EXIT_ON_CLOSE);
20 frame.setLayout(new BorderLayout());
21 frame.add(listScroll, BorderLayout.
  CENTER);
22 frame.setPreferredSize(new Dimension(500
  , 500));
23 frame.pack();
24 frame.setVisible(true);

```

Cela nous donne le résultat suivant à l'écran :

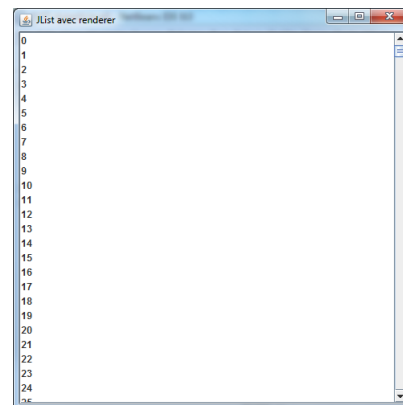


Figure 2 - Une bonne vieille JList d'antan.

Une fois la liste affichée, un petit tour par la console va nous dévoiler une information étonnante : la méthode *getListCellRendererComponent()* de notre *renderer*, celle qui permet de customiser le tampon pour un élément à un indice donné de la liste, a été appelée plus de 3000 fois ! Pourtant seuls les 26 premiers éléments de la liste sont visibles (on aperçoit un bout de la valeur 25).

En effet, la *JList* a besoin de calculer sa taille maximale de manière à permettre au *JScrollPane* d'afficher correctement ses barres de défilement. Le *renderer* est donc configuré successivement pour chacun des 3000 éléments de la liste pour déterminer cette dimension. Et ce n'est pas fini ! En effet, à

chaque redimensionnement de la fenêtre, cette méthode peut être appelée plus de 1000 fois ! Cette méthode est également appelée de très nombreuses fois à chaque défilement du contenu de la liste. . .

Ce n'est pas très efficace ! Imaginez ce qu'il en se-

### 3 JavaFX

Dans JavaFX, les composants qui sont destinés à afficher un grand nombre de valeurs ont été « virtualisés », c'est-à-dire qu'ils affichent une vue qui ne contient que les éléments qui sont visibles.

#### 3.1 Concepts de l'API Cell

Dans le cas d'une liste telle que celle ci-dessous par exemple, chaque ligne visible est représentée par une cellule. En fait, les lignes contenant des valeurs qui sont complètement en dehors de la zone d'affichage n'existent pas dans la vue. Le calcul de la hauteur totale de la liste est effectué en extrapolant les valeurs retournées par les cellules qui existent réellement et non plus sur l'intégralité des éléments de la liste. Il en va de même pour les composants qui peuvent défiler horizontalement. Il s'agit là de l'application d'un principe souvent ignoré de la programmation orientée objet : la réutilisation/le recyclage des objets existants !

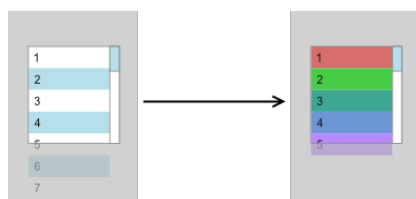


Figure 3 - Seules les parties visibles nous intéressent.

Si on fait défiler les données, les cellules qui deviennent invisibles seront récupérées, réutilisées et reconfigurées, comme l'était le *renderer* de Swing, de manière à afficher les nouveaux éléments qui apparaissent sur les lignes nouvellement visibles. Si la liste est agrandie (à l'écran) de nouvelles cellules seront créées pour afficher les lignes supplémentaires.



Figure 4 - Création et réutilisation de cellules lors du défilement.

Ici, nous voyons bien que seuls les éléments visibles sont affichés par des cellules :

rait avec un affichage beaucoup plus complexe dans la cellule ou avec un nombre beaucoup plus grand d'éléments qui nous seraient retournés, par exemple, par une requête sur une base de données ou un SIG.

- lorsque l'élément 6 devient visible, une nouvelle cellule orange est créée ; à ce stade, l'élément 1 est encore partiellement visible, la cellule rouge est donc toujours présente ;
- plus tard durant le défilement, la cellule rouge contenant l'élément 1 est retirée de l'affichage lorsque cet élément n'est plus visible. Cette cellule est conservée dans un pool de cellules ;
- la cellule rouge est ensuite réutilisée lorsque l'élément 7 devient visible à son tour ;
- la cellule vert clair contenant l'élément 2 disparaît à son tour lorsque cet élément n'est plus visible.

Bien sûr, il s'agit là du fonctionnement théorique du concept : des améliorations de performances sont encore possibles dans la gestion de la vue virtuelle. Ainsi, l'API Cell disponible dans les JDK 7 et 8 effectue des appels redondants lorsqu'un élément est modifié (même en dehors du champ de la vue) : toutes les cellules visibles sont mises à jour même si ce n'est pas nécessaire. Des correctifs de performances ont été implémentés et seront probablement intégrés lors de mises à jour ultérieures du JDK8.

#### 3.2 Mise en œuvre

Prenons le code suivant (nous reviendrons sur ce qu'il fait de manière plus détaillée ultérieurement) :

```

1 final ListView<Integer> list = new
  ListView<>();
2 for (int index = 0; index < 3000; index++
  ) {
3     list.getItems().add(index);
4 }
5 list.setCellFactory(new Callback<
  ListView<Integer>, ListCell<Integer>
  >() {
6     private int created;
7
8     @Override
9     public ListCell<Integer> call(
10        ListView<Integer> p) {
11        created++;
12        System.out.println("====Created
13        =====Created " + created);
14        return new ListCell<Integer>() {
15            private int called;
16
17            @Override
18            protected void updateItem(
19                Integer value, boolean
20                empty) {
21                final int cellIndex =
22                    created;

```

```

18         called++;
19         System.out.println("
           Called " + cellIndex
           + " " + called);
20         super.updateItem(value,
           empty);
21         final String text = (
           value == null ||
           empty) ? null :
           String.valueOf(value
           );
22         setText(text);
23     }
24 }
25 }
26 });
27 final StackPane root = new StackPane();
28 root.getChildren().add(list);
29 final Scene scene = new Scene(root);
30 primaryStage.setTitle("ListView");
31 primaryStage.setWidth(500);
32 primaryStage.setHeight(500);
33 primaryStage.setScene(scene);
34 primaryStage.show();

```

Ici, nous remplissons également une liste graphique avec 3000 éléments. Nous spécifions une fabrique (lien 93) à cellules sur la *ListView*. Dans cette fabrique, à chaque appel à la méthode *call()*, nous créons une nouvelle implémentation d'une cellule capable de transformer un *Integer* en *String* pour procéder à son affichage. Cette transformation s'effectue lorsque la méthode *updateItem()* est appelée. Il n'y a pas besoin de spécifier un *ScrollPane*, les composants JavaFX peuvent afficher des barres de scroll quand nécessaire.

Cela nous donne le résultat suivant à l'écran :

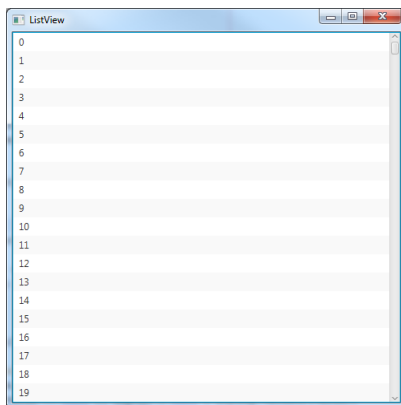


Figure 5 - Une *ListView* de nombres.

Ici, seuls les 20 premiers éléments sont visibles (la police et la hauteur des lignes dans JavaFX sont différentes de celles de Swing). Un petit tour sur la console nous montre que la méthode *call()* de la fabrique à cellules placée sur notre *ListView* a été appelée 21 fois environ (une cellule supplémentaire est créée pour la ligne affichant le nombre 20). La méthode *updateItem()* des cellules créées a été, quant à elle, appelée 2-3 fois en moyenne pour chaque cellule. On reste très loin des 3000+ appels dans la version Swing, même si ici, on a créé 21 objets cellule au lieu d'un seul objet *renderer*.

C'est lors de l'utilisation de la liste que cette manière de faire prend tout son sens : si nous faisons défiler le contenu de la liste ou si nous redimensionnons la fenêtre horizontalement, nous nous apercevons alors que seule la méthode *updateItem()* de chacune des 21 cellules existantes est appelée. Aucune nouvelle cellule n'est créée.

En fait la méthode *call()* de la fabrique à cellule n'est appelée que si nous agrandissons notre fenêtre verticalement : en effet pour chaque nouvelle ligne qui devient visible, une nouvelle cellule doit être générée.

Si on réduit au contraire la taille de la fenêtre, on se rend alors compte que la méthode *updateItem()* n'est appelée que sur les cellules qui sont visibles à l'écran.

Note : si on réduit la taille de la liste jusqu'à n'afficher qu'une seule ligne, les cellules non visibles sont toujours référencées en mémoire. Elles seront à nouveau utilisées si on fait défiler les données ou si on agrandit la fenêtre.

C'est un sacré gain de performances et de mémoire qui permet d'afficher des listes contenant un très très grand nombre d'éléments avec une rapidité d'affichage maximale ou encore sur des environnements restreints tels que des plateformes mobiles.

### 3.3 Contrôles virtualisés

Voici la liste de tous les contrôles actuels dits « virtualisés » de JavaFX et qui utilisent l'API Cell.

- *javafx.scene.control.ListView<V>* : une liste graphique.
- *javafx.scene.control.ComboBox<V>* : une boîte déroulante.
- *javafx.scene.control.TableView<V>* : une table affichant des données sur plusieurs colonnes.
- *javafx.scene.control.TreeView<V>* : un arbre graphique.
- *javafx.scene.control.TreeTableView<V>* : une table dont la première colonne contient un arbre. JDK8 ou supérieur.

On peut classer ces contrôles en deux grandes familles : ceux utilisant une liste d'objets et ceux utilisant une arborescence d'objets.

Les contrôles à base de liste :

- *ListView*
- *ComboBox*
- *TableView*

Les contrôles à base d'arbre :

- *TreeView*
- *TreeTableView*



### 3.4 ChoiceBox<V>

Le contrôle *javafx.scene.control.ChoiceBox<V>* ne fait pas partie des contrôles supportant l'API *Cell*. Ce contrôle est destiné à afficher une liste déroulante et ressemble à s'y méprendre à une *ComboBox*. Bien qu'acceptant n'importe quel type d'objet, la liste déroulante ne peut afficher que du texte simple sans aucune possibilité d'en changer la présentation ou d'ajouter des éléments graphiques ou riches. De plus, la boîte déroulante n'est pas éditable.

La propriété *convertir*, de type *StringConverter<V>*, permet de spécifier un convertisseur qui prendra un élément de type *V* en entrée et produira une *String* en sortie.

Dans la plupart des cas on préférera sans doute utiliser *ComboBox* à la place de cette classe, sauf

quand on doit montrer une courte liste de valeurs simples à l'utilisateur.

### 3.5 DatePicker

Le contrôle *javafx.scene.control.DatePicker*, ajouté dans le JavaFX 8, permet d'utiliser une boîte déroulante pour sélectionner une date dans un calendrier qui s'affiche dans un menu surgissant. Ce contrôle utilise également l'API *Cell* pour générer les cellules qui servent à l'affichage de chaque jour du calendrier.

Le programmeur peut utiliser sa propriété *dayCellFactory* pour fournir une fabrique à cellules permettant de customiser l'apparence de certains jours dans le calendrier, par exemple : afficher les anniversaires ou les fêtes ou jours fériés ou encore des rendez-vous.

## 4 L'API Cell

La classe de base de l'API Cell est la classe *javafx.scene.control.Cell<V>* où *V* est le type de l'élément présent dans le conteneur. Cette classe hérite du contrôle *javafx.scene.control.Labeled* qui est la classe parente, entre autres, de la classe *Label* et des diverses classes de boutons dans l'API SceneGraph JavaFX.

### 4.1 Classes

La classe *Cell<V>* a deux classes filles :

- *javafx.scene.control.IndexedCell<V>* - qui contient une propriété *index*. C'est cette classe qui est utilisée dans les contrôles « virtualisés » ;
- *javafx.scene.control.DateCell* - une classe de cellules utilisée par le contrôle *DatePicker*. JDK 8 ou supérieur.

La classe *IndexedCell<V>* est ensuite elle-même dérivée en classes destinées aux divers contrôles « virtualisés » :

- *javafx.scene.control.ListView<V>* - est destinée à être utilisée dans *ListView<V>* et *ComboBox<V>* ;
- *javafx.scene.control.TableCell<V, T>* - est destinée à être utilisée dans *TableView<V>*. Ici *T* est le type de l'objet affiché dans la colonne dans laquelle se trouve la cellule ;
- *javafx.scene.control.TableRow<V>* - est destinée à être utilisée dans *TableView<V>*. Cette classe est rarement utilisée sauf si le programmeur veut remplacer une ligne entière de la table ;
- *javafx.scene.control.TreeCell<V>* - est destinée à être utilisée dans *TreeView<V>* ;

- *javafx.scene.control.TreeTableCell<V, T>* - est destinée à être utilisée dans *TreeTableView<V>*. Ici *T* est le type de l'objet affiché dans la colonne dans laquelle se trouve la cellule. JDK8 ou supérieur ;
- *javafx.scene.control.TreeTableRow<V>* - est destinée à être utilisée dans *TreeTableView<V>*. Cette classe est rarement utilisée sauf si le programmeur veut remplacer une ligne entière de la table. JDK8 ou supérieur.

À partir de JavaFX 2.2, ces classes disposent d'une multitude de classes filles concrètes prêtes à l'emploi, spécialement destinées à afficher des champs de saisie texte, des cases à cocher, des barres de progression, ou même des boîtes déroulantes, etc.

Cela nous donne un graphe de hiérarchie de classes similaire à celui-ci :

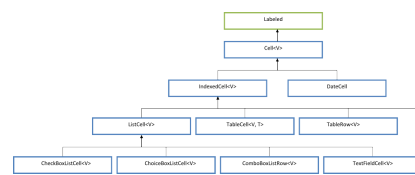


Figure 6 - Hiérarchie des classes.

### 4.2 Propriétés

Le fait d'hériter de la classe *Labeled*, signifie qu'une cellule partage plusieurs propriétés en commun avec le contrôle *Label* et les divers contrôles de boutons (*Button*, *CheckBox*, *RadioButton*, *ToggleButton*, etc.). Une cellule se manipule de manière similaire :

- la propriété *text* permet d'afficher du texte ;
- le texte dans la cellule peut être configuré au moyen de diverses propriétés liées :



- *ellipsisString* - le texte à utiliser en cas d'ellipse,
  - *font* - la police de caractères du texte,
  - *lineSpacing* - permet de spécifier l'espace entre les lignes en cas de texte multiligne. JDK8 ou supérieur,
  - *mnemonicParsing* - indique si l'API doit rechercher le caractère `_` comme indicateur de raccourci clavier,
  - *textAlignment* - permet de spécifier l'alignement du paragraphe en cas de texte multiligne,
  - *textFill* - permet de spécifier la couleur ou la texture du texte,
  - *textOverflow* - spécifie le type d'ellipse à utiliser quand le texte doit être coupé lors de l'affichage.
  - *underline* - permet de spécifier si le texte doit être souligné,
  - *wrapText* - indique si un texte trop long peut avoir un retour à la ligne;
  - la propriété *graphic* permet de spécifier un « graphique », un nœud quelconque permettant d'avoir un affichage riche :
    - une icône via une instance du contrôle *ImageView*,
    - un hyperlien via une instance du contrôle *HyperLink*,
    - un texte riche via une instance du contrôle *TextFlow*. JDK8 ou supérieur,
    - une apparence riche et interactive beaucoup plus complexe qu'un simple label en spécifiant n'importe quel nœud ou contrôle;
  - le positionnement du graphique par rapport au texte peut être spécifié via diverses propriétés :
    - *contentDisplay* - permet de choisir entre un affichage texte seul, graphique seul ou combinant les deux ainsi que la position relative de l'un par rapport à l'autre,
    - *graphicTextGap* - permet de spécifier l'espacement entre le texte et le graphique;
  - d'autres propriétés s'appliquent sur le couple texte + graphique :
    - *alignment* - l'alignement du couple à l'intérieur du label,
    - *labelPadding* - l'espacement tout autour du couple à l'intérieur du label;
  - les accesseurs CSS contrôlant l'aspect et le positionnement du texte et du graphique sont similaires à ceux des autres classes héritant de *Labeled*.
- De plus, la classe *Cell* contient des propriétés qui lui sont propres et qui permettent l'édition lorsqu'elle est utilisée dans certains contrôles :
- *editable* - indique s'il est possible d'éditer le contenu de la cellule;
  - *editing* - une propriété en lecture seule qui indique si le contenu de la cellule est en train d'être édité;
  - *empty* - une propriété en lecture seule qui indique si la cellule est vide et qu'il n'y a pas de données à visualiser (ex. : les lignes vides d'une table qui complètent l'affichage lorsque la table contient peu de données);
  - *item* - l'objet contenu dans la cellule. La valeur *null* est une valeur valide sans pour autant que la cellule soit considérée comme vide;
  - *selected* - une propriété en lecture seule qui indique si la cellule est sélectionnée.
- Enfin, les cellules héritant de la classe *IndexedCell* disposent également de la propriété :
- *index* - une propriété en lecture seule qui indique la position de la cellule dans le contrôle « virtualisé ».

Retrouvez la suite de l'article de **Fabrice Bouyé** en ligne : [lien 94](#)

# Qt



## Les dernières news Qt

# [Android] Apparition d'un style natif pour Android dans Qt 5.4

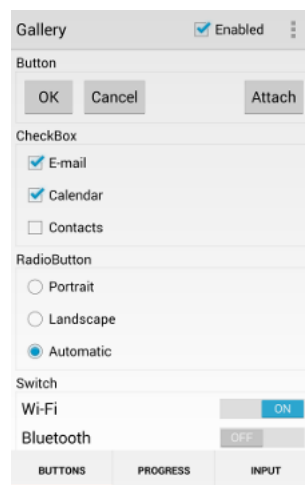
Comme vous l'avez peut-être entendu, Qt 5.4 (lien 95), qui sera prochainement disponible en version finale, introduit un nouveau style pour Android. Cet article se veut pragmatique, en exposant les nouveautés pour vos applications Android développées avec Qt.

### Qt Widgets

Il était par le passé possible d'avoir des applications reposant sur Qt Widgets ayant une apparence native grâce à Ministro (lien 96), un système installant les bibliothèques Qt au niveau du système Android. Dans Qt 5.4, une partie du code source de Ministro a été incorporée dans le module de plateforme pour Android. Cela permet aux applications Qt d'avoir une apparence native, même si Ministro n'est pas installé. Ceci étant, les applications voulant utiliser des fonctionnalités additionnelles proposées par Ministro pourront continuer à le faire. Pour résumer la situation, les applications utilisant Qt Widgets auront une apparence native, indépendamment de la façon dont Qt aura été distribué.

### Contrôles Qt Quick

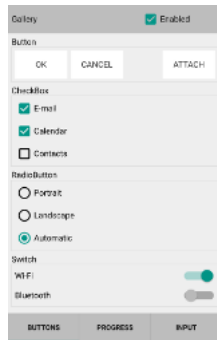
Une nouveauté majeure de Qt 5.4 est l'introduction d'un nouveau style pour les applications utilisant les contrôles Qt Quick. Vous pouvez en avoir un aperçu via la capture d'écran de l'application de démonstration, tournant sur un Nexus 5 sous Android KitKat 4.4. Par ailleurs, il est à noter que ce style Android nécessite Android 3.0 (API niveau 11) ou ultérieur. Sur des appareils plus anciens, un thème générique reposant sur QML sera utilisé à la place.



### Android 5.0

Les plates-formes mobiles évoluent à un rythme effréné. Alors que les équipes travaillaient dur pour fournir un style générique fonctionnant sur Android 3.0 et ultérieur, une nouvelle version majeure, à savoir Android 5.0 « Lollipop », fut introduite, ainsi qu'un nouveau thème, le « Material design » : lien 97. Ce nouveau thème introduit tellement de nouveautés qu'il n'a pas été possible de toutes les prendre en compte à temps.

Ainsi, le support du thème « Material » n'est malheureusement pas à un niveau satisfaisant pour le moment. Les applications reposant sur Qt 5.4.0 utiliseront donc le thème par défaut, à savoir Holo, sur Android 5.0. Les problèmes les plus visibles sont : des effets de transitions manquants, des couleurs de teinte manquantes, des effets de vague, ainsi que des animations indéfinies et/ou bloquantes (QTBUG-42520 (lien 98) et QTBUG-42644 (lien 99)).



Pour les plus curieux d'entre vous, qui ne peuvent pas attendre que ces défauts soient corrigés, vous

Commentez la news d'**Arnold Dumas** en ligne : [lien 100](#)

## Sortie de Qt 5.4

Qt 5.4 est dès à présent disponible, depuis la section des téléchargements de [qt.io](#), en même temps que Qt Creator 3.3 et qu'une mise à jour de Qt pour la création de systèmes sous Linux embarqué ou Android.

### Une offre Web renouvelée

HTML5 et les technologies dites « Web » occupent une place toujours plus importante, c'est pourquoi lors de ces dernières années l'offre Web de Qt a été complètement renouvelée. Le module Qt WebEngine ([lien 101](#)) est l'aboutissement d'un projet de recherche et développement de longue haleine, qui se trouve être une adaptation du moteur Web Chromium pour Qt. Ce moteur est maintenant pleinement supporté sur la plupart des plates-formes de bureau et embarquées. Qt WebEngine propose une API intuitive permettant de présenter du contenu Web dans des applications reposant sur Qt Widgets ou Qt Quick.

Le nouveau module Qt WebChannel ([lien 102](#)) permet une interaction facile entre QML/C++ et HTML/JavaScript. Cela permet la création d'applications hybrides faisant à la fois appel à Qt et aux technologies Web. La communication entre ces deux mondes se fait en exposant des QObjects dans un contexte Web. Ce module fonctionne non seulement avec Qt WebEngine, mais aussi avec n'importe quel moteur Web gérant les *Web sockets*.

Troisième nouveauté, Qt 5.4 introduit un support encore limité du nouveau module Qt WebView : [lien 103](#). Ce nouveau module offre une API volontairement limitée, permettant d'embarquer une vue Web, native pour le système d'exploitation sous-jacent, dans les cas où le Qt WebEngine n'est pas nécessaire ou bien quand son usage est impossible de par les restrictions imposées par le système d'exploitation. Dans Qt 5.4, le module Qt WebView est

pouvés dès à présent utiliser le thème « Material » en éditant le fichier `AndroidManifest.xml` comme suit :

```

1 <manifest ...>
2   <application ... android:theme="
3     @android:style/Theme.Material.
4     Light">
5   ...
6 </application>
7 </manifest>
```

Cette méthode peut aussi être utilisée pour utiliser la version claire ou foncée du thème Holo : il faudra alors utiliser les chaînes « `Theme.Holo.Light` » et « `Theme.Holo` » respectivement.

disponible sous iOS et Android.

Grâce au module Qt WebSockets ([lien 104](#)) introduit par Qt 5.3, Qt propose maintenant un support de qualité pour de nombreuses technologies Web de dernier cri, vous permettant d'interagir facilement avec ce type de contenu. Les modules Qt WebEngine et Qt WebView facilitent fortement l'intégration de contenu HTML5 au sein d'une application Qt, tandis que Qt WebChannel rend possible la communication entre Qt et HTML5, nécessaire dans le cas d'applications hybrides ; finalement, Qt WebSockets permet la communication entre Qt et différents services Web.

Qt 5.4 contient toujours l'ancien module Qt WebKit. Il est toujours supporté, mais, à partir de maintenant, il est considéré comme terminé et ne recevra donc aucune nouvelle fonctionnalité. Par ailleurs, ce module sera déprécié dans une prochaine version du *framework*, étant donné que Qt WebEngine propose toutes les fonctionnalités attendues. Dans la plupart des cas, la migration de Qt WebKit vers Qt WebEngine ne pose aucun souci. Si vous démarrez un projet faisant appel à des technologies Web, vous êtes invités à directement démarrer avec Qt WebEngine.

### Qt pour WinRT : une offre multiplateforme toujours plus riche

La seconde grosse nouveauté pour Qt 5.4 est la gestion complète de Windows Runtime : [lien 105](#). Qt pour Windows Runtime fit son apparition en qualité de Beta dans Qt 5.3 et est désormais mature. Avec Qt pour Windows Runtime, vous pouvez maintenant développer des applications pour le Windows Store et ainsi viser les Windows Phone (8.1 et ultérieur) ainsi que Windows (8.1 et ultérieur).

### Mise à jour de la pile graphique

Qt 5.4 apporte un certain nombre de nouvelles

fonctionnalités et d'améliorations dans le domaine des graphismes. Ainsi, cette nouvelle version introduit une gestion améliorée des écrans à haute densité pour les plates-formes de bureau. Ce support est toujours considéré comme expérimental. Si ce domaine vous intéresse, vous pouvez consulter la documentation à ce sujet : [lien 106](#).

L'utilisation d'OpenGL sous Windows était parfois à l'origine de problèmes, notamment du fait de l'absence de bons pilotes. Pour faciliter la résolution de ce problème, Qt est maintenant capable de dynamiquement changer d'implémentation OpenGL ([lien 107](#)) au démarrage de l'application : Qt choisira entre l'implémentation native d'OpenGL et ANGLE, qui fera ensuite appel à DirectX ou à un rendu logiciel en cas de besoin.

Une autre amélioration de la pile graphique ([lien 108](#)) consiste en l'introduction de la nouvelle classe QOpenGLWidget ([lien 109](#)), qui remplace la vieillissante classe QGLWidget de Qt 4. Cela permet de déprécier le module Qt OpenGL, étant donné que toutes ses fonctionnalités sont maintenant disponibles dans Qt Gui. QOpenGLContext ([lien 110](#)) est maintenant capable d'encapsuler un contexte natif existant. Vous pouvez utiliser la nouvelle classe QQuickRenderControl ([lien 111](#)) pour afficher une scène Qt Quick dans un tampon hors-champ. Pour plus de détails, consultez l'article sur ces nouvelles classes : [lien 112](#).

Enfin, Qt 5.4 contient une prévue technologique du nouveau module Qt Canvas3D ([lien 113](#)), qui implémente une API similaire à WebGL pour Qt Quick. Ce module facilite grandement l'utilisation de JavaScript faisant appel à WebGL avec du Qt Quick. Qt 5.4 contient tellement de nouveautés qu'il est impossible de toutes les détailler. Avant de poursuivre la lecture de cette actualité, vous êtes invité à visualiser cette vidéo : [lien 114](#)

#### **Autres nouvelles fonctionnalités**

Une quantité importante de nouvelles fonctionnalités est disponible dans cette nouvelle version de Qt, vous trouverez ci-joint un descriptif des plus significatifs : [lien 115](#).

Qt supporte maintenant le Bluetooth basse consommation sous Linux via BlueZ : [lien 116](#). Le support d'autres plates-formes fera son apparition

dans les prochaines versions du *framework*. Le Bluetooth basse consommation rend possible la communication avec de nombreux appareils Bluetooth, notamment les objets et vêtements connectés.

Du côté d'Android, les contrôles Qt Quick ont maintenant une apparence native ([lien 117](#)), les paquets de déploiement sont plus petits et les temps de démarrage sont réduits. Les dernières versions d'iOS et d'OS X sont maintenant gérées, ainsi que Xcode 6 ou encore le nouveau processus de signature de code, obligatoire pour publier sur l'App Store. De gros efforts ont été consentis pour corriger les différentes erreurs relatives au nouveau style d'OS X 10.10 (Yosemite).

Le support des machines à états en QML ([lien 118](#)) fait son apparition via l'import Qt-Qml.StateMachine. La classe QStorageInfo, petit dernier au sein du module Qt Core, permet l'obtention d'informations concernant les périphériques et volumes montés. Qt 5.4 est livré avec une nouvelle version de Qt Creator, à savoir Qt Creator 3.3 ([lien 119](#)). Pour plus de détails sur cette nouvelle mouture, vous pouvez consulter l'article dédié : [lien 120](#).

#### **Introduction de la LGPL v3**

Comme annoncé précédemment ([lien 121](#)), la version open source de Qt 5.4 est maintenant disponible sous les termes de la LGPLv3 : [lien 122](#). Cette nouvelle licence permet à la Qt Company d'offrir des composants à forte valeur ajoutée pour tous les utilisateurs de Qt sans faire de concessions du côté commercial.

Dans Qt 5.4, certains modules sont uniquement disponibles sous GPL/LGPLv3 ou sous licence commerciale : Qt WebEngine ainsi que la prévue technologique de Qt WebView, mais aussi Qt Canvas 3D. Le style Android est uniquement disponible sous LGPLv3 ou sous une licence commerciale. Vous trouverez plus d'informations à ce sujet sur le nouveau site de Qt : [lien 123](#).

#### **Remerciements à la communauté**

Qt 5.4 introduit un nombre important de nouvelles fonctionnalités et d'améliorations majeures. Certaines d'entre elles n'auraient jamais pu voir le jour sans les contributions de la communauté et d'entreprises partenaires, qui contribuent à Qt sans être affiliées à la Qt Company.

*Commentez la news d'Arnold Dumas en ligne : [lien 124](#)*

# Sortie de Qt Creator 3.3.0

Qt Creator 3.3.0 est dès à présent disponible. Cette nouvelle version vient avec de nombreuses nouvelles fonctionnalités ainsi qu'avec un nombre important de bogues corrigés.

La liste des nouvelles fonctionnalités a déjà été détaillée dans le précédent article traitant de la version Beta : [lien 125](#). Pour la version finale, une vidéo de présentation est maintenant disponible : [lien 126](#).

Parmi les autres nouvelles fonctionnalités, on peut mentionner le support du moteur de compilation Gradle pour Android, une action de réorganisation du code permettant de migrer vers la nouvelle syntaxe de connexion d'un signal à un slot, apparue avec Qt 5, ou encore le support des projets CMake pour des appareils nus. Pour une liste détaillée des changements, vous êtes invité à consulter le fichier des changements : [lien 127](#).

Qt Creator 3.3.0 fait partie des installeurs de Qt

5.4.0, disponible depuis aujourd'hui : [lien 128](#).

Les deux sont téléchargeables depuis la page des téléchargements de qt.io : [lien 129](#). Merci de faire remonter les bogues que vous pourriez rencontrer sur l'application de gestion des bogues : [lien 130](#). Vous pouvez aussi discuter avec les équipes de développements sur IRC, canal #qt-creator chez irc.freenode.net, et/ou via la liste de diffusion de Qt Creator : [lien 131](#).

**Note :** Qt Creator 3.3 n'est plus compilable avec Qt 4, il faut désormais utiliser Qt 5.3.1 ou plus récent pour compiler l'EDI. Bien sûr, cela n'affecte en rien les versions de Qt que vous pouvez utiliser pour vos développements : les projets reposant sur Qt 4 sont toujours totalement pris en charge. Si vous souhaitez utiliser des add-on pour Qt Creator, vous devrez cependant les rendre compilables avec Qt 5.

*Commentez la news d'**Arnold Dumas** en ligne : [lien 132](#)*



# CRM

## Les derniers tutoriels et articles

# Développer un batch en Apex dans Salesforce

Comment développer, tester et planifier un batch dans Salesforce.com ?

## 1 Introduction

Il peut arriver (dans Salesforce.com ou d'autres systèmes d'information) de devoir exécuter un traitement asynchrone sur des données (de petite ou grande quantité) pour les archiver, les nettoyer ou pour un autre besoin.

Le batch répond tout à fait à ce besoin puisqu'il s'agit ni plus ni moins d'une tâche CRON que l'on retrouve sur d'autres systèmes informatisés plus communs tels que Unix ou Windows.

Avant d'aller plus loin dans cet article, je vous propose quelques rappels sur les termes techniques qui vont être utilisés :

- Salesforce.com ([lien 133](#)) est un CRM (Customer Relationship Manager) orienté SaaS (Software as a Service : [lien 134](#)) et PaaS (Platform as a Service : [lien 135](#)) permettant la gestion des relations clients d'une entreprise ;
- Apex, est un langage de programmation côté serveur permettant de modifier la logique métier et le traitement des données ;

- SOQL (Salesforce Object Query Language) est le langage permettant d'effectuer des requêtes dans la base de données de Salesforce.com.

Maintenant, place au scénario de ce tutoriel : nous souhaitons mettre à jour toutes les nuits, le nombre de points des comptes en fonction des événements commerciaux auxquels ils ont participé.

C'est-à-dire que si un compte participe à un événement commercial, cela lui rapporte un certain nombre de points suivant son niveau (bronze, argent ou or) et l'année de l'événement en question.

Un compte peut donc participer à un ou plusieurs événements commerciaux et un événement commercial peut être suivi par un ou plusieurs comptes.

Voici donc le schéma du modèle de données que je propose :

Il peut y avoir des notions que vous ne comprenez pas dans ce tutoriel, mais pour éviter de faire de la redondance, je vous invite à vous reporter vers la section « Ressources utiles ».

## 2 Prérequis

Avant toute chose, vous devez disposer d'une instance afin de pouvoir suivre ce tutoriel. Si ce n'est pas le cas, je vous invite à vous rendre à cette adresse : [lien 136](#). Salesforce met à disposition plusieurs types d'instances qui proposent différentes fonctionnalités et donc à différents prix (prix par licence utilisateur par mois). La seule instance que vous pourrez généreusement obtenir de la part de Salesforce est celle de développement (celle que je vous propose via le lien ci-dessus) qui, comme son nom l'indique, sert à tester et développer des services. Je

vous laisse juger par vous-même la grille tarifaire ([lien 137](#)) sur les diverses instances que propose le CRM.

Vous aurez également besoin d'un environnement de développement pour écrire votre code, vous pouvez utiliser celui fourni par Salesforce ou alors vous procurer celui compatible avec Eclipse et qui s'appelle Force.com IDE (je l'utiliserai tout au long de ce tutoriel). Pour ce dernier, rendez-vous sur l'Eclipse MarketPlace et installez-le.

## 3 Définition et fonctionnement d'un batch

Un batch est une classe Apex qui lorsqu'elle est appelée se place à la fin d'une file d'attente (queue

en anglais) pour qu'elle puisse être exécutée lorsque



cela sera possible (en fonction du nombre de traitements à effectuer).

Vous pouvez apercevoir vos planifications dans Salesforce.com en vous rendant à Setup => Monitoring => Scheduled Jobs et la file d'attente dans Setup => Monitoring => Apex Jobs.

Les batchs traitent par défaut des lots de 200 enregistrements (mais il est possible de spécifier une autre valeur entre 1 et 10 000 lors de son exécution). Ce qui signifie que si nous voulons des lots de 200 enregistrements et que nous avons 500 enregistrements à traiter, nous appellerons donc trois fois le batch. Il faut savoir qu'à chaque appel du batch, les governor limits sont réinitialisées, cela permet une plus grande souplesse sur les limites qu'impose Salesforce.com.

Par limites, on peut y trouver :

- 100 requêtes SOQL lors d'une exécution synchrone (contre 200 en asynchrone) ;

## 4 Développer un batch

Lors de la création d'un batch, il faut respecter certaines conditions pour que celui-ci soit valide, la classe doit au minimum :

- implémenter l'interface Database.Batchable ;
- contenir les méthodes start(), execute() et finish().

Dans Salesforce.com, chaque lot d'enregistrements d'un batch est considéré comme une transaction discrète (discrete transaction en anglais), cela vous permet, si vous spécifiez l'implémentation de l'interface Database.Stateful, de garder l'état de vos variables durant le traitement de tous les lots, elles ne seront jamais réinitialisées, cela peut être pratique pour des variables booléennes ou pour des map contenant des enregistrements.

Autre point également important à connaître, si un lot d'enregistrements échoue pour une raison ou une autre, les autres lots ne seront pas impactés et suivront leur déroulement prévu.

Maintenant, commençons par créer notre batch avec son minimum vital :

```

1  global class
   BatchEventsAccountPointsCalculation
   implements Database.Batchable<
   sObject>, Database.Stateful{
2
3   public String query;
4
5   global Database.QueryLocator start(
   Database.BatchableContext bc){
6
7       return Database.getQueryLocator(
   query);
8   }
9
10  global void execute(Database.
   BatchableContext BC, List<
   sObject> scope){
11

```

- 10 000 enregistrements maximum retournés lors de l'exécution de la requête SOQL de la méthode DataBase.getQueryLocator() ;
- 120 secondes maximum avant un temps mort lors d'un appel à un service web ou d'une requête HTTP ;
- 10 secondes de temps CPU allouées lors d'une transaction APEX (lors d'un traitement côté serveur) ;
- 50 appels à des méthodes annotées future lors d'une transaction APEX.

Il existe plein d'autres limites qu'il peut être appréciable de connaître pour développer.

Je vous partage le lien vers la documentation concernant ces limites dans la partie VIII Ressources utiles.

```

12   }
13
14   global void finish(Database.
   BatchableContext BC){
15
16   }
17 }

```

La méthode start() est utilisée pour charger les données sur lesquelles nous effectuerons le traitement, dans notre cas, il s'agit des comptes.

Nous devons retourner la méthode Database.getQueryLocator() en lui fournissant en paramètre, la requête SOQL (Salesforce Object Query Language) permettant de requêter les comptes au format String.

La méthode execute() quant à elle, permet d'effectuer tout le traitement nécessaire sur les enregistrements que l'on vient de récupérer à l'aide de la requête dans la méthode start() en l'occurrence, des comptes.

Passons maintenant à la dernière méthode, finish(). Elle est appelée en dernière lorsque le traitement de execute() est terminé.

Nous pouvons utiliser cette méthode pour envoyer des mails de confirmation du traitement à des utilisateurs, appeler un autre batch ou encore autre chose.

Afin de faciliter au maximum la maintenance du code, il est très préférable que lorsque vous avez besoin d'utiliser des valeurs fixes (qui ne dépendent pas d'enregistrements) comme du texte ou des nombres, d'utiliser des labels personnalisés (custom labels) dans Salesforce.com. Cela facilite la maintenance et dans le cas où par exemple, je remplace le niveau des comptes silver en autre chose, un custom label permet une modification rapide et propre et ne nécessite pas une nouvelle mise en production (change

set, exécution des classes de test, etc.) vu qu'il n'y a pas de modification du code.

Dans ce tutoriel, pour savoir quels points attribuer aux comptes, je dois savoir de quel type ils sont (argent, gold ou platinum) et comme ce sont des valeurs fixes, j'ai créé des labels que j'utilise dans le batch (au travers de variables en les récupérant avec System.Label.nom\_de\_mon\_label).

Voici l'algorithme que je propose.

Il nous faut récupérer les informations des comptes (avec au minimum leur identifiant et leur niveau) ainsi que tous les événements auxquels ils ont participé.

Une fois que nous avons les comptes, nous itérons dessus dans un premier temps, puis sur leurs événements pour leur attribuer les points suivant leur niveau.

Voici le résultat :

```

1  /** Batch qui calcule les points de
2  participation à des événements
3  commerciaux des comptes */
4  global class
5  BatchEventsAccountPointsCalculation
6  implements Database.Batchable<
7  sObject>, Database.Stateful{
8
9  public String query;
10 private Integer currentYear;
11 private List<Account> accountList;
12 private String silverLabel = System.
13 Label.Label_silver;
14 private String goldLabel = System.
15 Label.Label_gold;
16 private String platinumLabel = System
17 Label.Label_platinum;
18
19 global Database.QueryLocator start(
20 Database.BatchableContext bc){
21
22     this.accountList = new List<Account>
23     ();
24
25     this.currentYear = Date.today().year
26     ();
27
28     this.query = 'SELECT Id, Name,
29     Level__c, Points__c, (SELECT Id,
30     Account__c,
31     Events_Business_Points__r.
32     Silver_level_points__c,
33     Events_Business_Points__r.
34     Gold_level_points__c,
35     Events_Business_Points__r.
36     Platinum_level_points__c,
37     Events_Business_Points__r.
38     Year__c FROM Participations__r
39     WHERE Events_Business_Points__r.
40     Year__c = :currentYear) FROM
41     Account WHERE Level__c != NULL';
42
43     return Database.getQueryLocator(this
44     .query);
45 }

```

```

22 global void execute(Database.
23 BatchableContext BC, List<sObject>
24 scope){
25
26     // Pour chaque compte
27     for(sObject aScope : scope){
28
29         Account anAccount = (Account)
30         aScope;
31
32         Decimal nbPoints = 0;
33
34         // Pour chaque participation du
35         compte
36         for(Participation__c
37         aParticipation : anAccount.
38         Participations__r){
39
40             // Si compte silver
41             if(anAccount.Level__c.equals(
42             silverLabel) &&
43             aParticipation.
44             Events_Business_Points__r.
45             Silver_level_points__c !=
46             null){
47                 nbPoints += aParticipation.
48                 Events_Business_Points__r.
49                 Silver_level_points__c;
50             }
51
52             // Sinon si compte gold
53             else if(anAccount.Level__c.
54             equals(goldLabel) &&
55             aParticipation.
56             Events_Business_Points__r.
57             Gold_level_points__c != null
58             ){
59                 nbPoints += aParticipation.
60                 Events_Business_Points__r.
61                 Gold_level_points__c;
62             }
63
64             // Sinon si compte platinum
65             else if(anAccount.Level__c.
66             equals(platinumLabel) &&
67             aParticipation.
68             Events_Business_Points__r.
69             Platinum_level_points__c !=
70             null){
71                 nbPoints += aParticipation.
72                 Events_Business_Points__r.
73                 Platinum_level_points__c;
74             }
75
76             anAccount.Points__c = nbPoints;
77
78             this.accountList.add(anAccount);
79         }
80
81         // Mise a jour des comptes
82         update this.accountList;
83     }
84
85     global void finish(Database.
86     BatchableContext BC){
87
88     }
89 }

```

## 5 Tester un batch

Comme tout développement mérite sa classe de test, nous sommes obligés pour nous assurer que notre batch fonctionne correctement, de le tester.

Pour cela, nous recréons un jeu de données qui va nous servir à le tester.

Il faut le plus possible éviter d'utiliser les données de l'instance.

Nous aurons donc besoin de créer :

- des comptes;
- des événements commerciaux;
- des points pour les événements commerciaux par année;
- des participations des comptes à des événements commerciaux.

Comme il peut parfois y avoir besoin d'effectuer plusieurs tests pour valider une classe Apex (tester avec plusieurs contextes), je recommande de concevoir une méthode d'initialisation (`init()`) dans notre classe de test afin d'initialiser notre environnement de test et de factoriser le code et son éventuelle maintenance.

Grâce à ça, vous n'aurez seulement qu'à appeler cette méthode dans les tests et vos données seront prêtes.

Maintenant que nous avons nos données, nous devons appeler notre batch pour qu'il s'exécute, mais nous devons le lancer à l'intérieur des balises `Test.startTest()` et `Test.stopTest()` afin que les limites imposées par Salesforce.com (je vous en ai parlé un peu plus haut) puissent être prises en compte.

Une fois le batch exécuté, nous récupérons nos comptes à partir de Salesforce.com avec leurs nouveaux points pour vérifier qu'ils ont bien le bon nombre et que notre batch est par conséquent, bien développé et éviter tout problème une fois mis en production avec de vraies données :

```

1 / Classe de test du batch
   BatchEventsAccountPointsCalculation
   /
2 @isTest
3 private class
   BatchEventsAccountPointsCalculation_TEST
4 {
5     static List<Account> accountList;
6     static List<Business_Event__c>
       businessEventList;
7     static List<Events_Business_Points__c>
       eventBusinessPointList;
8     static List<Participation__c>
       participationList;
9     static String silverLabel = System.
       Label.Label_silver;
10    static String goldLabel = System.Label
       .Label_gold;
11    static String platinumLabel = System.
       Label.Label_platinum;

```

```

12    static Date todayDate;
13
14    static void init(){
15        todayDate = Date.today();
16
17        /** Account **/
18        accountList = new List<Account>();
19        accountList.add(new Account(
20            Name = 'Test name silver',
21            Level__c = silverLabel
22        ));
23        accountList.add(new Account(
24            Name = 'Test name gold',
25            Level__c = goldLabel
26        ));
27        accountList.add(new Account(
28            Name = 'Test name platinum',
29            Level__c = platinumLabel
30        ));
31
32        insert accountList;
33
34        /** Business_Event__c **/
35        businessEventList = new List<
36            Business_Event__c>();
37        businessEventList.add(new
38            Business_Event__c(
39                Name = 'Salesforce1 World Tour
40                Paris',
41                Place__c = 'Paris'
42            ));
43        businessEventList.add(new
44            Business_Event__c(
45                Name = 'Dreamforce',
46                Place__c = 'Los Angeles'
47            ));
48        businessEventList.add(new
49            Business_Event__c(
50                Name = 'Salesforce1 World Tour
51                London',
52                Place__c = 'London'
53            ));
54        insert businessEventList;
55
56        /** Events_Business_Points__c **/
57        eventBusinessPointList = new List<
58            Events_Business_Points__c>();
59        eventBusinessPointList.add(new
60            Events_Business_Points__c(
61                Business_Event__c =
62                businessEventList[0].Id,
63                Year__c = todayDate.year()-1,
64                Silver_level_points__c = 10,
65                Gold_level_points__c = 15,
66                Platinum_level_points__c = 30
67            ));
68        eventBusinessPointList.add(new
69            Events_Business_Points__c(
70                Business_Event__c =
71                businessEventList[0].Id,
72                Year__c = todayDate.year(),
73                Silver_level_points__c = 20,
74                Gold_level_points__c = 35,
75                Platinum_level_points__c = 40
76            ));
77        eventBusinessPointList.add(new
78            Events_Business_Points__c(
79                Business_Event__c =
80                businessEventList[1].Id,

```

```

69     Year__c = todayDate.year()-1,
70     Silver_level_points__c = 100,
71     Gold_level_points__c = 150,
72     Platinum_level_points__c = 300
73   ));
74   eventBusinessPointList.add(new
75     Events_Business_Points__c(
76     Business_Event__c =
77       businessEventList[1].Id,
78     Year__c = todayDate.year(),
79     Silver_level_points__c = 150,
80     Gold_level_points__c = 300,
81     Platinum_level_points__c = 450
82   ));
83   eventBusinessPointList.add(new
84     Events_Business_Points__c(
85     Business_Event__c =
86       businessEventList[2].Id,
87     Year__c = todayDate.year(),
88     Silver_level_points__c = 120,
89     Gold_level_points__c = 270,
90     Platinum_level_points__c = 430
91   ));
92   insert eventBusinessPointList;
93
94   /** Participation__c **/
95   participationList = new List<
96     Participation__c>();
97   participationList.add(new
98     Participation__c(
99     Account__c = accountList[0].Id,
100    Events_Business_Points__c =
101      eventBusinessPointList[0].Id
102   ));
103   participationList.add(new
104     Participation__c(
105     Account__c = accountList[0].Id,
106     Events_Business_Points__c =
107       eventBusinessPointList[1].Id
108   ));
109   participationList.add(new
110     Participation__c(
111     Account__c = accountList[1].Id,
112     Events_Business_Points__c =
113       eventBusinessPointList[4].Id
114   ));
115   participationList.add(new
116     Participation__c(
117     Account__c = accountList[2].Id,
118     Events_Business_Points__c =
119       eventBusinessPointList[2].Id
120   ));
121   participationList.add(new
122     Participation__c(
123     Account__c = accountList[2].Id,
124     Events_Business_Points__c =
125       eventBusinessPointList[4].Id
126   ));
127   insert participationList;
128 }
129
130 /** Teste les points du compte **/
131 static testMethod void
132   testAccountPoint() {
133   init();
134   Test.startTest();
135
136   Database.executeBatch(new
137     BatchEventsAccountPointsCalculation
138     ());
139   Test.stopTest();
140
141   // Récupération des comptes dans
142   // Salesforce
143   List<Account> newAccountList = [
144     SELECT Id, Name, Points__c
145     FROM Account
146     WHERE Id IN :accountList
147   ];
148
149   // Verification des points
150   System.assertEquals(
151     eventBusinessPointList[1].
152     Silver_level_points__c,
153     newAccountList[0].Points__c)
154   ;
155   System.assertEquals(
156     eventBusinessPointList[4].
157     Gold_level_points__c,
158     newAccountList[1].Points__c)
159   ;
160   System.assertEquals((
161     eventBusinessPointList[1].
162     Platinum_level_points__c +
163     eventBusinessPointList[4].
164     Platinum_level_points__c),
165     newAccountList[2].Points__c)
166   ;
167 }

```

## 6 Planifier l'exécution d'un batch

Développer un batch, c'est bien, mais il nous faut maintenant créer une classe qui va nous permettre de l'exécuter.

Pour ce faire, la classe doit obligatoirement implémenter l'interface `Schedulable`, mais seulement comporter l'exécution du batch en utilisant la méthode `Database.executeBatch()`.

Cette méthode attend deux paramètres :

- une instance du batch ;
- (optionnel) le nombre d'enregistrements maximum par lot (entre 1 et 10 000) que nous voulons, si ce n'est pas spécifié, la valeur de 200 est prise par défaut.

```

1 global class
  ScheduleBatchEventsAccountPointsCalc
  implements Schedulable{
2
3  global void execute(
  SchedulableContext ctx) {
4
5      Database.executeBatch(new
  BatchEventsAccountPointsCalculations
  ());
6
7  }
8 }

```

Comme toujours, tout développement doit comporter sa classe de test. Mais ici, elle est très simple, vous allez le voir.

Nous faisons appel à la méthode `System.schedule()` qui attend trois paramètres :

- un nom pour le job dans Salesforce.com ;
- la fréquence d'exécution du batch ;
- une instance de la classe qui exécute le batch.

Le tout, entre les balises `Test.startTest()` et `Test.stopTest()` bien entendu.

```

1 @isTest(seeAllData=false)
2 private class
  ScheduleBatchEventsAccountPointsCalc_TEST
  {
3
4  static testMethod void
  ScheduleBatchSalesGrowthCalculation_TEST
  () {
5      Test.startTest();
6
7      System.schedule('
  ScheduleBatchEventsAccountPointsCalc
  ', '0 0 3 * * ?', new
  ScheduleBatchEventsAccountPointsCalc
  ());
8
9      Test.stopTest();
10 }

```

```
11 }
```

Il ne nous reste qu'une étape dans la planification d'un batch.

Cela se fait en une ligne de code à exécuter soit dans la console du développeur soit dans la section Exécution anonyme dans le plugin Force.com IDE dans Eclipse.

Nous exécutons la même méthode que celle utilisée dans notre classe de test vue précédemment `System.schedule()`.

L'expression de la planification comprend six paramètres obligatoires et un optionnel :

- les secondes ;
- les minutes ;
- les heures ;
- le jour du mois ;
- le mois ;
- le jour de la semaine ;
- l'année.

Voilà les valeurs disponibles pour l'expression de la planification :

Je ne vais pas redétailler les trois paramètres qu'elle utilise. Dans mon cas, je souhaite que le batch s'exécute toutes les nuits à 3 h 00.

La ligne de code ressemble donc à ceci :

```

1 System.schedule('
  ScheduleBatchEventsAccountPointsCalc
  ', '0 0 3 * * ?', new
  ScheduleBatchEventsAccountPointsCalc
  ());

```

La fréquence d'exécution du batch (2e paramètre) correspond à l'expression CRON que l'on retrouve dans n'importe quel système Unix et Windows.

## 7 Conclusion

Nous arrivons à la fin de ce tutoriel, j'espère vous avoir correctement expliqué comment développer, planifier et tester un batch dans Salesforce.com.

Cela peut paraître long et ennuyeux lorsque l'on développe dans Salesforce.com (développement + test), mais cela s'avère nécessaire pour s'assurer que ce que l'on fait fonctionne correctement parce qu'une fois qu'une donnée en production devient périmée suite à un faux comportement, cela est trop tard et ce sera toujours le développeur qui se fera taper sur les doigts, donc il est important de livrer quelque chose de sain.

## 8 Ressources utiles

- Apex Batch Processing : [lien 141](#) ;
- Using Batch Apex : [lien 142](#) ;

Comme je suis quelqu'un de sympa, je vous mets à disposition (ici : [lien 138](#)), toutes les sources qui ont été nécessaires à la réalisation de cet article, à savoir :

- le batch ;
- la classe de test du batch ;
- la classe d'exécution du batch ;
- la classe de test de la classe d'exécution du batch.

Elles sont également disponibles sur mon compte Bitbucket ([lien 139](#)) à cette adresse : [lien 140](#).

- Apex Scheduler : [lien 143](#) ; [mits](#) : [lien 144](#) ;
- Understanding Execution Governors and Li- [Tester son code Apex](#) : [lien 145](#).

Retrouvez l'article de **Aurélien Laval** en ligne : [lien 146](#)



# Liste des liens

## Page 2

- lien 1 : ... <https://github.com/Microsoft/TypeScript/pull/824>
- lien 2 : ... <https://github.com/Microsoft/TypeScript/issues/805>
- lien 3 : ... <https://docs.angularjs.org/api/ng/service/\protect\T1\textdollarhttp#usage>

## Page 5

- lien 4 : ... <http://yahiko.developpez.com/tutoriels/typescript-union-types/>
- lien 5 : ... <http://www.typescriptlang.org/>

## Page 9

- lien 6 : ... <http://nodejs.org/download/>
- lien 7 : ... <http://www.microsoft.com/france/visual-studio/>
- lien 8 : ... <https://www.jetbrains.com/webstorm/>
- lien 9 : ... <http://typecsdev.com/>
- lien 10 : ... <http://typescript.io/>
- lien 11 : ... <http://jsfiddle.net/>
- lien 12 : ... <http://www.typescriptlang.org/Playground/>
- lien 13 : ... <https://github.com/Microsoft/TypeScript>

## Page 10

- lien 14 : ... <https://github.com/>
- lien 15 : ... <http://www.githubarchive.org/>
- lien 16 : ... <http://www.babylonjs.com/>
- lien 17 : ... <http://blogs.msdn.com/b/eternalcoding/archive/2014/04/28/why-we-decided-to-move-from-plain-javascript-to-typescript-for-babylon-js.aspx>

## Page 11

- lien 18 : ... <http://www.typescriptlang.org/Playground/>
- lien 19 : ... <http://www.typescriptlang.org/Handbook>
- lien 20 : ... <http://www.typescriptlang.org/Content/TypeScript%20Language%20Specification.pdf>
- lien 21 : ... <http://asmjs.org/>
- lien 22 : ... <https://gist.github.com/darthapo/3916195>

## Page 12

- lien 23 : ... <http://yahiko.developpez.com/tutoriels/introduction-typescript/#L11-2>
- lien 24 : ... <http://yahiko.developpez.com/tutoriels/introduction-typescript/>
- lien 25 : ... <http://www.developpez.net/forums/d1477435/webmasters-developpement-web/javascript-ajax-typescript-dart/typescript/introduction-langage-typescript-javascriptpp/>

## Page 13

- lien 26 : ... <http://www.iancgbell.clara.net/elite/text/>
- lien 27 : ... <http://yahiko.co.nf/universeGenerator/>
- lien 28 : ... <http://www.developpez.net/forums/d1470523/applications/developpement-2d-3d-jeux/generation-aleatoire-d-univers/>
- lien 29 : ... [http://setosa.io/markov/index.html#%7B%22tm%22%3A\[\[0.9%2C0.1\]%2C\[0.5%2C0.5\]\]%7D](http://setosa.io/markov/index.html#%7B%22tm%22%3A[[0.9%2C0.1]%2C[0.5%2C0.5]]%7D)

## Page 14

- lien 30 : ... <http://www.di.ens.fr/~lelarge/proba09/ENSmarkov.pdf>

## Page 16

- lien 31 : ... <http://typescript.io/wEixPeu0Qg>

## Page 17

- lien 32 : ... <http://yahiko.co.nf/NameGenerator/>
- lien 33 : ... <https://github.com/yahiko00/NameGenerator>
- lien 34 : ... [http://fr.wikipedia.org/wiki/Processus\\_de\\_décision\\_markovien](http://fr.wikipedia.org/wiki/Processus_de_décision_markovien)
- lien 35 : ... <http://yahiko.developpez.com/tutoriels/generation-aleatoire-nom/>

## Page 18

- lien 36 : ... <http://gdcvault.com/play/1019219/Triangulation-A-Schizophrenic-Approach-to>

**Page 20**

**lien 37** : ... [http://en.wikipedia.org/wiki/Civilization\\_\(video\\_game\)](http://en.wikipedia.org/wiki/Civilization_(video_game))

**Page 22**

**lien 38** : ... <http://noorshaker.com/CutTheRope.html#content-inner-2>

**lien 39** : ... <http://www.purplepaw.com/2008/07/ludi-the-electronic-game-designer/>

**lien 40** : ... <http://aigamedev.com/open/interview/evolution-in-cityconquest/>

**lien 41** : ... <http://www.pcgamer.com/2013/03/19/richard-garriott-game-designers-suck/>

**Page 24**

**lien 42** : ... <http://tcuvelier.developpez.com/tutoriels/jeux/modelisation-et-optimisation-decisions-dans-conception-jeux/01-introduction/>

**Page 25**

**lien 43** : ... <http://www.framablog.org/index.php/post/2013/10/07/emmabuntus-plus-qu-une-distribution-linux>

**lien 44** : ... <http://linux.developpez.com/tutoriels/debuter-installation/guide-linux-distribution/>

**lien 45** : ... <http://distrowatch.com/table.php?distribution=emmabuntus>

**lien 46** : ... [http://emmabuntus.sourceforge.net/mediawiki/index.php/Linux\\_Essentiel\\_33](http://emmabuntus.sourceforge.net/mediawiki/index.php/Linux_Essentiel_33)

**lien 47** : ... [http://emmabuntus.sourceforge.net/mediawiki/index.php/Linux\\_Pratique\\_Hors\\_Serie\\_28](http://emmabuntus.sourceforge.net/mediawiki/index.php/Linux_Pratique_Hors_Serie_28)

**lien 48** : ... [http://emmabuntus.sourceforge.net/mediawiki/index.php/L\\_age\\_de\\_faire\\_Mars\\_2014](http://emmabuntus.sourceforge.net/mediawiki/index.php/L_age_de_faire_Mars_2014)

**lien 49** : ... [http://emmabuntus.sourceforge.net/mediawiki/index.php/Terra\\_Eco\\_Septembre\\_2014](http://emmabuntus.sourceforge.net/mediawiki/index.php/Terra_Eco_Septembre_2014)

**lien 50** : ... <http://montpel-libre.fr/spip.php?page=recherche&recherche=emmabunt%C3%BCs>

**lien 51** : ... [http://emmabuntus.sourceforge.net/mediawiki/index.php/Reemploi\\_Linux](http://emmabuntus.sourceforge.net/mediawiki/index.php/Reemploi_Linux)

**lien 52** : ... <http://www.ordi-solidaire.fr/content/7-suites-logicielles>

**lien 53** : ... <http://www.framasoft.net/article5216.html>

**lien 54** : ... <http://distrowatch.com/search.php?category=Old+Computers>

**Page 26**

**lien 55** : ... <http://mikedafunk.over-blog.org/article-post-installation-lubuntu-14-04-123930366.html>

**lien 56** : ... <http://linux.developpez.com/tutoriels/HandyLinux-Debian-sans-se-prendre-la-tete/>

**lien 57** : ... <http://emmabuntus.sourceforge.net/mediawiki/index.php/Download>

**lien 58** : ... <http://www.developpez.net/forums/d1480748/systemes/linux/distributions/ubuntu/emmabuntues-ubuntu-emmaues/>

**Page 27**

**lien 59** : ... <http://www.ganttproject.biz/download>

**Page 42**

**lien 60** : ... <http://vviale.developpez.com/tutoriels/ganttproject/>

**Page 43**

**lien 61** : ... <http://sourceforge.net/projects/projectlibre/files/ProjectLibre/>

**Page 54**

**lien 62** : ... <http://vviale.developpez.com/tutoriels/projectlibre/>

**Page 56**

**lien 63** : ... <http://fr.wikipedia.org/wiki/Endianness>

**Page 57**

**lien 64** : ... <http://perl.developpez.com/documentations/en/5.20.0/Encode.html>

**lien 65** : ... [http://www.perlmonks.org/?node\\_id=644786](http://www.perlmonks.org/?node_id=644786)

**Page 58**

**lien 66** : ... <http://search.cpan.org/perldoc?Devel::Peek>

**lien 67** : ... <http://blogs.perl.org/users/aristotle/2011/08/utf8-flag.html>

**Page 59**

**lien 68** : ... <http://search.cpan.org/perldoc?Template::Alloy>

**lien 69** : ... <http://search.cpan.org/perldoc?HTML::Template::Compiled>

**Page 60**

**lien 70** : ... <http://unicode.org/faq/normalization.html>

lien 71 : ... <http://www.w3.org/TR/2004/REC-xml-20040204/#sec-guessing-no-ext-info>

#### Page 61

lien 72 : ... <http://www.w3.org/International/tutorials/tutorial-char-enc/%20>

lien 73 : ... [http://en.wikibooks.org/wiki/Perl\\_Programming/Unicode\\_UTF-8](http://en.wikibooks.org/wiki/Perl_Programming/Unicode_UTF-8)

lien 74 : ... <http://perldoc.perl.org/perlunitut.html>

lien 75 : ... <http://live.gnome.org/Gucharmap>

lien 76 : ... <http://perlgeek.de/blog-en/perl-tips/utf8-dump.html>

lien 77 : ... <http://www.freebsd.org/cgi/man.cgi?query=hexdump>

lien 78 : ... <http://www.gnu.org/software/libiconv/documentation/libiconv/iconv.1.html>

lien 79 : ... <http://perl.developpez.com/tutoriels/perl/encodage-caracteres/>

#### Page 67

lien 80 : ... <http://vviale.developpez.com/tutoriels/openoffice-libreoffice/apprendre-creer-depliant-brochure/>

#### Page 70

lien 81 : ... <http://openoffice-libreoffice.developpez.com/tutoriels/openoffice-libreoffice/calc-tableur-mysql/>

#### Page 71

lien 82 : ... <http://openjdk.java.net/jeps/158>

lien 83 : ... <http://openjdk.java.net/jeps/165>

lien 84 : ... <http://openjdk.java.net/jeps/214>

lien 85 : ... <http://openjdk.java.net/jeps/213>

lien 86 : ... <http://openjdk.java.net/jeps/201>

lien 87 : ... <http://openjdk.java.net/jeps/220>

lien 88 : ... <http://www.developpez.net/forums/d1482715/java/general-java/oracle-presente-nouvelles-fonctionnalites-java-9-a/>

#### Page 72

lien 89 : ... <http://openjdk.java.net/jeps/223>

lien 90 : ... <http://www.developpez.net/forums/d1484965/java/general-java/jdk-1-9-n-existera/>

#### Page 73

lien 91 : ... <http://www.developpez.net/forums/d1482715-2/java/general-java/oracle-presente-nouvelles-fonctionnalites-java-9-a/#post8063255>

#### Page 74

lien 92 : ... [http://fr.wikipedia.org/wiki/Poids-mouche\\_\(patron\\_de\\_conception\)](http://fr.wikipedia.org/wiki/Poids-mouche_(patron_de_conception))

#### Page 76

lien 93 : ... [http://fr.wikipedia.org/wiki/Fabrique\\_abstraite\\_\(patron\\_de\\_conception\)](http://fr.wikipedia.org/wiki/Fabrique_abstraite_(patron_de_conception))

#### Page 78

lien 94 : ... <http://fabrice-bouye.developpez.com/tutoriels/javafxcell/>

#### Page 79

lien 95 : ... <http://qt.developpez.com/actu/76403/Sortie-de-Qt-5-4-Beta-avec-une-meilleure-gestion-des-ecrans-a-haute-resolution-et-du-Bluetooth-basse-consommation/>

lien 96 : ... <https://necessitas.kde.org/necessitas/ministro.php>

lien 97 : ... <https://developer.android.com/design/material/index.html>

lien 98 : ... <https://bugreports.qt-project.org/browse/QTBUG-42520>

lien 99 : ... <https://bugreports.qt-project.org/browse/QTBUG-42644>

#### Page 80

lien 100 : ... <http://www.developpez.net/forums/d1486150/c-cpp/bibliotheques/qt/platformes/mobiles/android-apparition-d-style-natif-android-qt-5-4-a/>

lien 101 : ... <http://doc.qt.io/qt-5/qtwebengine-index.html>

lien 102 : ... <http://doc.qt.io/qt-5/qtwebchannel-index.html>

lien 103 : ... <http://doc.qt.io/qt-5.4/qtwebview-index.html>

lien 104 : ... <http://doc.qt.io/qt-5/qtwebsockets-index.html>

lien 105 : ... <http://doc.qt.io/qt-5/winrt-support.html>

#### Page 81

lien 106 : ... <http://doc.qt.io/qt-5/highdpi.html>  
 lien 107 : ... <http://blog.qt.digia.com/blog/2014/11/27/qt-weekly-21-dynamic-opengl-implementation-loading-in-qt-5-4/>  
 lien 108 : ... <http://blog.qt.digia.com/blog/2014/11/20/qt-weekly-20-completing-the-offering-qopenglwindow-and-qrasterwindow/>  
 lien 109 : ... <http://doc.qt.io/qt-5/qopenglwidget.html>  
 lien 110 : ... <http://doc.qt.io/qt-5/qopenglcontext.html>  
 lien 111 : ... <http://doc.qt.io/qt-5/qquickrendercontrol.html>  
 lien 112 : ... <http://blog.qt.digia.com/blog/2014/09/10/qt-weekly-19-qopenglwidget/>  
 lien 113 : ... <http://doc.qt.io/qt-5.4/qtcanvas3d-index.html>  
 lien 114 : ... [https://www.youtube.com/watch?feature=player\\_embedded&v=wMs1pSZMnGO](https://www.youtube.com/watch?feature=player_embedded&v=wMs1pSZMnGO)  
 lien 115 : ... <http://qt-project.org/wiki/New-Features-in-Qt-5.4>  
 lien 116 : ... <http://doc.qt.io/qt-5/qtbluetooth-le-overview.html>  
 lien 117 : ... <http://doc.qt.io/qt-5/qtquick-controls-qmlmodule.html>  
 lien 118 : ... <http://doc.qt.io/qt-5/qmlstatemachine.html>  
 lien 119 : ... <http://doc.qt.io/qtcreator>  
 lien 120 : ... <http://www.developpez.net/forums/d1479378/c-cpp/bibliotheques/qt/edi/qt-creator/sortie-qt-creator-3-3-0-a/#post8064199>  
 lien 121 : ... <http://qt.developpez.com/actu/74292/Nouvelle-licence-pour-Qt-le-framework-sera-egalement-disponible-sous-la-LGPL-3-avec-de-nouveaux-modules-libres/>  
 lien 122 : ... <http://opensource.org/licenses/lgpl-3.0.html>  
 lien 123 : ... <http://www.qt.io/licensing/>  
 lien 124 : ... <http://www.developpez.net/forums/d1468332/c-cpp/bibliotheques/qt/sortie-qt-5-4-a/#post8064176>

#### Page 82

lien 125 : ... <http://qt.developpez.com/actu/76864/Sortie-de-Qt-Creator-3-3-Beta-avec-de-nouvelles-fonctionnalites-d-organisation-de-l-espace-de-travail-et-la-gestion-de-Gradle-pour-Android/>  
 lien 126 : ... [https://www.youtube.com/watch?feature=player\\_embedded&v=8IatD3iq6yY](https://www.youtube.com/watch?feature=player_embedded&v=8IatD3iq6yY)  
 lien 127 : ... <https://qt.gitorious.org/qt-creator/qt-creator/source/3.3:dist/changes-3.3.0>  
 lien 128 : ... <http://www.developpez.net/forums/d1468332/c-cpp/bibliotheques/qt/sortie-qt-5-4-a/#post8064176>  
 lien 129 : ... <http://www.qt.io/download>  
 lien 130 : ... <https://bugreports.qt-project.org/>  
 lien 131 : ... <http://lists.qt-project.org/mailman/listinfo/qt-creator>  
 lien 132 : ... <http://www.developpez.net/forums/d1479378/c-cpp/bibliotheques/qt/edi/qt-creator/sortie-qt-creator-3-3-0-a/#post8064199>

#### Page 83

lien 133 : ... <https://www.salesforce.com/fr/company/>  
 lien 134 : ... [http://fr.wikipedia.org/wiki/Logiciel\\_en\\_tant\\_que\\_service](http://fr.wikipedia.org/wiki/Logiciel_en_tant_que_service)  
 lien 135 : ... [http://fr.wikipedia.org/wiki/Plate-forme\\_en\\_tant\\_que\\_service](http://fr.wikipedia.org/wiki/Plate-forme_en_tant_que_service)  
 lien 136 : ... <https://developer.salesforce.com/>  
 lien 137 : ... <http://www.salesforce.com/fr/crm/editions-pricing.jsp>

#### Page 88

lien 138 : ... [http://aurelien-laval.developpez.com/tutoriels/salesforce/developper-un-batch-apex/fichiers/sources\\_tutoriel\\_salesforce\\_5.rar](http://aurelien-laval.developpez.com/tutoriels/salesforce/developper-un-batch-apex/fichiers/sources_tutoriel_salesforce_5.rar)  
 lien 139 : ... <https://bitbucket.org/AurelienLaval/>  
 lien 140 : ... <https://bitbucket.org/AurelienLaval/d-veloppeur-un-batch-apex-dans-salesforce>  
 lien 141 : ... [http://www.salesforce.com/us/developer/docs/apex\\_workbook/Content/apex\\_batch\\_intro.htm](http://www.salesforce.com/us/developer/docs/apex_workbook/Content/apex_batch_intro.htm)  
 lien 142 : ... [http://www.salesforce.com/us/developer/docs/apexcode/Content/apex\\_batch\\_interface.htm](http://www.salesforce.com/us/developer/docs/apexcode/Content/apex_batch_interface.htm)

#### Page 89

lien 143 : ... [https://www.salesforce.com/us/developer/docs/apexcode/Content/apex\\_scheduler.htm](https://www.salesforce.com/us/developer/docs/apexcode/Content/apex_scheduler.htm)  
 lien 144 : ... [https://www.salesforce.com/us/developer/docs/apexcode/Content/apex\\_gov\\_limits.htm](https://www.salesforce.com/us/developer/docs/apexcode/Content/apex_gov_limits.htm)  
 lien 145 : ... <http://aurelien-laval.developpez.com/tutoriels/salesforce/tester-son-code-apex/>  
 lien 146 : ... <http://aurelien-laval.developpez.com/tutoriels/salesforce/developper-un-batch-apex/>