



# Developpez

*Le Mag*

Édition de décembre – janvier 2015/2016

Numéro 61

Magazine en ligne gratuit

Diffusion de copies conformes à l'original autorisée

Réalisation : Alexandre Pottiez

Rédaction : la rédaction de Developpez

Contact : magazine@redaction-developpez.com

## Sommaire

C++	Page	2
Access	Page	9
Delphi	Page	20
Python	Page	26
LabVIEW	Page	43
NoSQL	Page	51
Solutions d'Entreprise	Page	52
Libres & Open Source	Page	53

## Éditorial

Le nouveau numéro du magazine est arrivé pour réchauffer vos longues nuits d'hiver. Retrouvez vos technologies favorites pour toujours plus de plaisir et de découverte.

La rédaction

## Article C++



### Utiliser CMake pour compiler un projet

CMake est un outil puissant pour gérer la construction de votre projet. Ce tutoriel vous guide dans l'utilisation de CMake.

par [Alexandre Laurent](#)

Page 2



## Article Python

### Concepts Python avancés

Il vous est sûrement arrivé de vous dire que si tous les modules/packages étaient stockés de manière centralisée, alors cela simplifierait grandement vos déploiements. Eh bien la fondation Python a eu la même réflexion, et a commencé à

travailler pour vous.

par [Alexandre Galode](#)

Page 26

# C++



## Les derniers tutoriels et articles

# Utiliser CMake pour compiler un projet

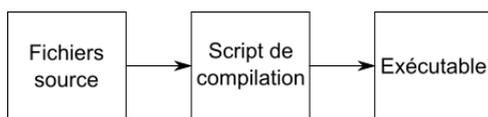
CMake est un outil puissant pour gérer la construction de votre projet. Ce tutoriel vous guide dans l'utilisation de CMake.

## 1 CMake

### 1.1 Introduction

CMake est un outil open source et gratuit permettant de gérer la compilation d'un projet.

Si nous prenons le modèle classique de compilation, nous avons :



Les **fichiers source** peuvent être des fichiers en C, C++, Java ou tout autre langage. Ceux-ci seront traités par le **script de compilation** qui appellera le compilateur (ou tout autre outil) pour créer l'**exécutable** final.

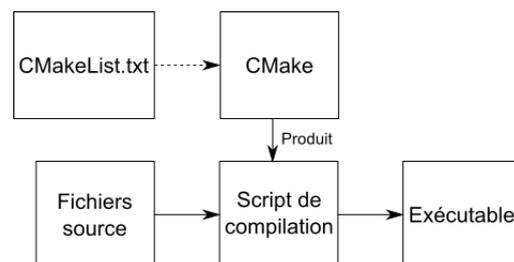
Le script de compilation peut avoir plusieurs formes suivant le projet et l'éditeur utilisé :

- un Makefile ;
- un fichier de projet Code : :Blocks ;
- un fichier de projet Microsoft Visual Studio ;
- un fichier de projet Eclipse ;
- ...

Toutefois, ce modèle possède quelques limites. En effet, si vous démarrez avec un projet Microsoft Visual Studio et que vous devez compiler sous Linux, vous serez obligé de recommencer la configuration du projet afin qu'il soit compilable sur le nouveau système. Cela peut aussi se produire, si vous commencez le projet sous Linux et que vous voulez faire un exécutable Windows, ou Mac OS X. Votre script de compilation dépend de votre système et/ou de votre chaîne de compilation. Ce problème existe pour les projets communautaires. En effet, les projets open source seront sûrement compilés par de nombreuses personnes qui utilisent des logiciels différents. Par exemple, le premier veut compiler sous Linux avec GCC, le second sous Windows avec Microsoft Visual Studio et un troisième avec une autre configuration.

Il est évident que gérer toutes les configurations serait une perte de temps et apporterait des complications inutiles.

C'est pourquoi CMake a été créé. En effet, CMake va résoudre cette problématique en créant les scripts de compilation à l'aide d'un fichier de configuration générique. Voici le nouveau modèle de compilation :



Cette fois, **CMake**, grâce au fichier **CMakeLists.txt**, va produire le script de compilation permettant la création de l'exécutable. Le fichier CMakeLists.txt est indépendant de la plateforme. Il décrit comment compiler le projet à l'aide d'informations comme : le langage utilisé, les fichiers à compiler, les dépendances (externes ou comme sous-projet). Ainsi CMake va pouvoir produire le script de compilation adéquat pour votre machine et votre projet.



Bien entendu, pour résoudre cette problématique, il existe différents logiciels similaires à CMake : autotools, scons, Jam...

### 1.2 Installation

CMake s'installe simplement : par le gestionnaire de paquets pour les distributions Linux, par un installateur pour Windows, ou par un fichier dmg pour MAC OS X téléchargeables directement sur le site officiel : [lien 1](#).



Pour Windows, je conseille vivement de rajouter le chemin d'installation dans la variable d'environnement **PATH**. Cela est proposé durant l'installation de CMake.



Bien entendu, pour compiler un projet, il ne suffit pas d'avoir CMake, il faut aussi un compilateur ou un EDI.

### 1.3 Fonctionnement

Avant de compiler un projet, vous devez générer le script de compilation à l'aide de CMake. Pour cela, vous devez lancer CMake dans le répertoire où se trouve le fichier CMakeLists.txt.

Pour générer le script de compilation, CMake utilise un générateur. Donc pour pouvoir générer le script de compilation que nous souhaitons avoir, nous devons indiquer le générateur à utiliser.



Les générateurs pour les compilateurs et IDE les plus courants sont fournis avec CMake.

### 1.4 Compiler un projet utilisant CMake

Il arrive souvent que l'on tombe sur des projets open source que nous devons compiler nous-mêmes afin de pouvoir les utiliser. Les projets CMake se reconnaissent grâce à la présence du fichier CMakeLists.txt (généralement à la racine du projet).

Voyons comment compiler ces projets.

#### 1.4.a Utilisation en ligne de commande

CMake peut être utilisé à travers l'invite de commande (ou terminal). Il suffit de taper :

```
cmake -G générateur
```

« générateur » correspond à l'un des générateurs proposés par CMake. Vous devez prendre celui correspondant à l'environnement avec lequel vous voulez compiler.



Vous pouvez obtenir la liste des générateurs avec la commande :

```
1 cmake --help
```

Si tout se passe correctement, les fichiers pour compiler le projet ont été générés. Sinon, lisez ce que CMake a affiché pour voir où il a bloqué et tentez de corriger suivant les messages.

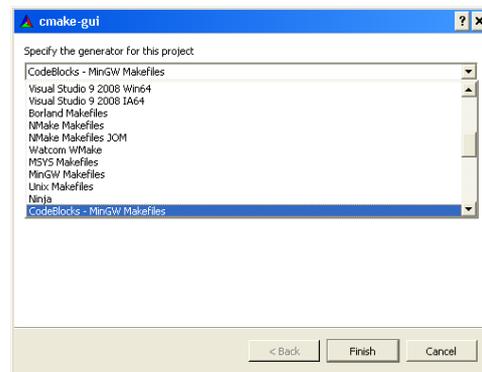
#### 1.4.b Utilisation par l'interface graphique

CMake peut aussi être utilisé à travers une interface graphique.

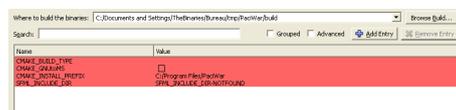
La première étape est de remplir les chemins « Where is the code source » (Où se trouve le source code) et « Where to build the binaries » (Où compiler les binaires). Pour le premier, vous devez indiquer le dossier contenant le CMakeLists.txt, le second doit indiquer où les fichiers générés seront placés. Ce répertoire sera l'emplacement de travail de votre compilateur ou EDI.



Ensuite, vous pouvez cliquer sur le bouton « Configure » pour lancer la configuration de CMake. Immédiatement (et aussi, car c'est votre premier lancement), il vous demandera le générateur à utiliser.



En validant la fenêtre de configuration, CMake va lancer la configuration du projet. Certains projets nécessitent de spécifier des chemins vers les bibliothèques, ou encore, des options de construction. Autant que possible, le CMakeLists.txt devrait être construit pour que CMake soit indépendant. Lorsque la configuration ne peut se finir sans un apport supplémentaire de la part de l'utilisateur, le bouton « Generate » restera grisé et l'interface permettra l'édition des variables utilisées dans le fichier CMakeLists.txt.



La couleur rouge pour les variables n'indique pas nécessairement que la configuration est fautive. Elle indique simplement que les variables sont nouvelles ou ont été modifiées lors de la dernière configuration.

Finalement, une fois la configuration finie, il suffit de cliquer sur le bouton « Generate » pour générer les fichiers permettant la compilation ou l'ouverture du projet dans un EDI.

## 1.5 Création du projet

Maintenant, nous allons voir comment créer un fichier CMakeLists.txt afin que CMake puisse générer les fichiers appropriés pour compiler notre projet.

### 1.5.a CMakeLists.txt

Comme nous l'avons vu, le fichier CMakeLists.txt est la pièce maîtresse pour compiler notre projet. Ce fichier décrit comment compiler le projet et permet donc à CMake de générer les scripts de compilation pour construire le projet.

#### 1.5.a.a Syntaxe

Le fichier CMakeLists.txt possède sa propre syntaxe. Celle-ci possède deux particularités :

- les commentaires sont spécifiés avec un crochillon '#';
- tout est fonction.

Il y a certes des tests conditionnels (if()/else()/elseif()), mais ils s'utilisent comme des fonctions. De même pour les variables, elles existent, mais vous devez les définir à travers la fonction set().



Le nom des fonctions n'est pas sensible à la casse (toutefois, on conseillera de toujours utiliser la même écriture tout au long du fichier). Par contre, le nom des variables l'est.

Toutes les variables sont de type chaîne de caractères. Il est aussi possible d'avoir des listes, mais celles-ci ne sont que des chaînes de caractères avec un format spécifique (plusieurs valeurs séparées par des points virgule ';').

Pour une description complète du langage, libre à vous de lire la documentation officielle : [lien 2](#).

#### 1.5.a.b Composition

Le CMakeLists.txt doit décrire votre projet. Pour cela, il faudra lui indiquer les éléments suivants :

- la version minimale pour lire le fichier CMakeLists.txt (cmake\_minimum\_required()) : [lien 3](#);
- le nom du projet (project()) : [lien 4](#);
- les dépendances à ajouter (target\_include\_directories() (lien 5)/target\_link\_libraries() (lien 6));
- le binaire à produire (add\_executable() (lien 7)/add\_library() (lien 8)).



Les fichiers à compiler sont donnés lors de l'appel à la fonction add\_executable()/add\_library().

### 1.5.a.c Exemple sur un projet simple

Pour nous entraîner, prenons un projet simple. C'est un projet en C, n'ayant aucune dépendance, juste trois fichiers source. Voici notre arborescence :

```

1 .
2 └─ CMakeLists.txt
3 └─ hello.c
4 └─ hello.h
5 └─ main.c

```

Vous pouvez télécharger le projet ici : [lien 9](#).

Voici le CMakeLists.txt correspondant :

```

1 # Nous voulons un cmake "récent" pour
   utiliser les dernières fonctionnalités
2 cmake_minimum_required(VERSION 3.0)
3
4 # Notre projet est étiqueté hello
5 project(hello)
6
7 # Crée des variables avec les fichiers à
   compiler
8 set(SRCS
9     main.c
10    hello.c
11    )
12
13 set(HEADERS
14    hello.h
15    )
16
17 # On indique que l'on veut un exécutable
   "hello" compilé à partir des
   fichiers décrits par les variables
   SRCS et HEADERS
18 add_executable(hello ${SRCS} ${HEADERS})

```

Et c'est tout ! Ce projet peut facilement compiler sous Linux, Windows, MAC OS et ce, peu importe le compilateur ou l'EDI que vous utilisez.

### 1.5.a.d Exemple sur un projet avec une bibliothèque en sous-projet

Ce second projet, toujours en C, crée d'abord une bibliothèque « libhello », puis un programme hello utilisant la fonction définie dans la bibliothèque. Voici l'arborescence du projet :

```

1 .
2 └─ CMakeLists.txt
3 └─ libhello
4   └─ CMakeLists.txt
5     └─ hello.c
6       └─ hello.h
7 └─ main.c

```

Notre projet est décomposé en deux modules :

- la bibliothèque libhello;
- le programme principal.

Vous pouvez télécharger le projet ici : [lien 10](#).

### 1.5.a.d.a La bibliothèque

Le CMakeLists.txt de la bibliothèque reprend tous les éléments que nous avons déjà étudiés, mais au lieu de générer un exécutable, c'est une bibliothèque que nous voulons. Pour l'indiquer, nous remplaçons `add_executable()` par `add_library()` :

```

1 # Nous voulons un cmake "récent" pour
  utiliser les dernières fonctionnalités
2 cmake_minimum_required(VERSION 3.0)
3
4 # Notre projet est étiqueté libhello
5 project(libhello)
6
7 # Crée des variables avec les fichiers à
  compiler
8 set(SRCS
9     hello.c
10    )
11
12 set(HEADERS
13     hello.h
14    )
15
16 add_library(hello ${SRCS} ${HEADERS})
    
```



Le nom du binaire indiqué à `add_library()` est « hello », car selon le système, CMake rajoutera « lib » comme préfixe afin de suivre les conventions du système cible.

### 1.5.a.d.b Le programme principal

Les particularités de ce nouveau CMakeLists.txt sont :

- l'inclusion d'un sous-projet pour compiler la bibliothèque (`add_subdirectory()`) : lien 11 ;
- la liaison de la bibliothèque avec le programme (`target_link_libraries()`) : lien 12.

```

1 # Nous voulons un cmake "récent" pour
  utiliser les dernières fonctionnalités
2 cmake_minimum_required(VERSION 3.0)
3
4 # Notre projet est étiqueté hello
5 project(hello)
6
7 # On inclut notre bibliothèque dans le
  processus de CMake
8 add_subdirectory(libhello)
9
10 # Crée des variables avec les fichiers à
  compiler
11 set(SRCS
12     main.c
13    )
14
15 # Notre exécutable
16 add_executable(main ${SRCS})
17
18 # Et pour que l'exécutable fonctionne,
  il faut lui indiquer la bibliothèque
  dont il dépend
19 target_link_libraries(main hello)
    
```

### 1.5.a.d.c Remarques

Si vous souhaitez simplement écrire `#include "hello.h"` dans le fichier `main.c`, vous pouvez rajouter `target_include_directory(libhello PUBLIC .)` à la fin du fichier `CMakeLists.txt` du dossier `libhello`.

### 1.5.a.e Exemple sur un projet avec une bibliothèque externe

Ce second projet, toujours en C, utilise la bibliothèque `libpng`. Voici l'architecture de notre projet :

```

1 .
2 └─ CMakeLists.txt
3 └─ main.c
    
```

Vous pouvez télécharger le projet ici : lien 13.

Dans le fichier `CMakeLists.txt` nous devons indiquer la dépendance à la bibliothèque `libpng`. Pour cela, CMake doit scanner différents dossiers (qui dépendent de la plateforme) pour savoir si les fichiers nécessaires sont présents ou non. Heureusement, nous n'avons pas à le faire manuellement. Il suffit d'utiliser la fonction `find_package()` : lien 14. La commande n'est pas magique. Elle va chercher dans les dossiers d'installation de CMake, un fichier `FindNOM_DE_LA_BIBLIOTHÈQUE.cmake` et le lire. Dans celui-ci, de simples fonctions et tests cherchant à savoir si la bibliothèque est installée et où. À la fin, plusieurs variables seront définies :

- `NOM_DE_LA_BIBLIOTHÈQUE_FOUND` : indique si oui ou non la bibliothèque a été trouvée ;
- `NON_DE_LA_BIBLIOTHÈQUE_INCLUDE_DIR/NOM` : le répertoire à inclure pour trouver les fichiers d'entêtes ;
- `NOM_DE_LA_BIBLIOTHÈQUE_LIBRARY/NOM_DE` : les bibliothèques à ajouter à l'édition de liens.



Certains modules définissent des variables supplémentaires. Il est donc conseillé de lire les commentaires du début du fichier du module.

Voici donc le CMakeFiles.txt de ce projet :

```

1 # Nous voulons un cmake "récent" pour
  utiliser les dernières fonctionnalités
2 cmake_minimum_required(VERSION 3.0)
3
4 # Notre projet est étiqueté hello
5 project(helloSDL)
6
7 # Crée des variables avec les fichiers à
  compiler
8 set(SRCS
9     main.c
10    )
11
12 # Notre exécutable
13 add_executable(main ${SRCS})
14
    
```

```

15 # Recherche la dépendance externe
16 find_package (PNG)
17 if (PNG_FOUND)
18     # Une fois la dépendance trouvée, nous
19     l'incluons au projet
20     target_include_directories(main PUBLIC
21     ${PNG_INCLUDE_DIR})
22     target_link_libraries (main ${
23     PNG_LIBRARY})
24 else ()
25     # Sinon, nous affichons un message
26     message(FATAL_ERROR "libpng not found
27     ")
28 endif ()
  
```



Il est possible de rechercher une version spécifique avec la syntaxe : `find_package(Boost 1.57.0)`.

### 1.5.a.f En vrac

#### 1.5.a.f.a Activation du C++11/C++14

Sur certains compilateurs, le support du C++11/C++14/... doit être activé en ajoutant une option supplémentaire lors de la compilation. La fonction appropriée pour ce faire est : `target_compile_features()` : lien 15.

Vous pouvez l'utiliser ainsi, notamment, pour vérifier si le compilateur supporte les `nullptr` :

```

1 # Cette ligne doit être placée après les
2   add_executable/add_library
3 target_compile_features(hello PUBLIC
4   cxx_nullptr)
  
```

Vous pouvez trouver la liste des fonctionnalités du C++ vérifiables à travers la fonction `target_compile_features()` dans la documentation : lien 16.

#### 1.5.a.f.b Copie de fichiers

La fonction `install()` (lien 17) permet de copier des fichiers :

```

1 install(TARGETS hello RUNTIME
2   DESTINATION .)
  
```

Il est aussi possible de copier des fichiers quelconques (données du projet) :

```

1 install(FILES fichier DESTINATION .)
  
```

ou encore un dossier :

```

1 install(DIRECTORY source DESTINATION .)
  
```



Chaque fichier peut nécessiter des actions particulières, il est important de lire la documentation pour trouver les arguments les plus appropriés.

### 1.5.a.f.c Passer des variables de CMake à votre code source

Par exemple, vous souhaitez afficher dans votre programme la version de celui-ci, ou encore, des informations de configuration du projet, déterminées par CMake lors de sa configuration. Pour cela, vous devez préparer un fichier qui servira de motif :

```

1 #define VERSION_MAJOR "
2   @HELLO_VERSION_MAJOR@"
  
```

Ensuite, dans votre `CMakeLists.txt` vous indiquez à CMake qu'il faut lire ce fichier et le transformer :

```

1 project(hello VERSION 1.0.1)
2
3 configure_file(${
4   CMAKE_CURRENT_SOURCE_DIR}/src/
5   version.h.in ${
6   CMAKE_CURRENT_SOURCE_DIR}/version.h)
  
```



Ici, nous utilisons les arguments de la fonction `project()` (lien 18) afin de définir la version de notre projet. À partir de cette information, CMake va définir les variables : `<NOM-DU-PROJET>_VERSION_MAJOR`, `<NOM-DU-PROJET>_VERSION_MINOR`, `<NOM-DU-PROJET>_VERSION_PATCH` (entre autres).

La fonction `configure_file()` (lien 19) lira le fichier indiqué en premier argument, cherchera les éléments englobés par des arobases et les remplacera par le contenu de la variable correspondant au nom trouvé entre les arobases. Le résultat (le fichier transformé) sera écrit dans le fichier indiqué en second argument.

Ce dernier pourra être inclus et utilisé dans votre code source.

## 1.6 Projet avec Qt

Bien que les projets Qt soient souvent compilés avec `qmake`, la solution de construction de projet associée à Qt, il est tout à fait possible de compiler son projet avec CMake.

### 1.6.a Vérifier la présence de Qt

CMake vous permet de vérifier la présence de la bibliothèque et d'obtenir les fichiers à lier à votre projet avec la fonction `find_package()` (lien 20) :

```

1 find_package(Qt5 COMPONENTS Widgets
2   OpenGL)
  
```

Dans cet exemple, nous souhaitons obtenir les modules « Widgets » et « OpenGL ». Par la suite, vous devrez utiliser « `Qt5::Widgets` » et «

Qt5 : « OpenGL » pour lier les bibliothèques de Qt à votre projet.

Une seconde syntaxe permet d'écrire :

```
1 find_package(Qt5Widgets)
```

Elle ne présente pas de différence avec la première.

### 1.6.b Exécution du moc

Une des particularités de la compilation d'un projet Qt est son utilisation du moc. CMake propose un mécanisme très simple pour définir lorsqu'un projet nécessite l'utilisation du moc :

```
1 # Indique d'utiliser moc lorsque né
  # cessaire.
2 set(CMAKE_AUTOMOC ON)
3 # Indique de chercher les entêtes dans
  # les répertoires de compilation.
4 # Cela est nécessaire, car les fichiers
  # du moc seront générés dans ces
  # dossiers.
5 set(CMAKE_INCLUDE_CURRENT_DIR ON)
```

### 1.6.c Utiliser des fichiers d'interface utilisateur (.ui)

Les fichiers d'interface utilisateur (.ui) de Qt doivent être transformés avant de pouvoir les inclure dans le binaire. Pour ce faire il faut utiliser la fonction `qt5_wrap_ui()` :

```
1 set(UIS interface1.ui interface2.ui)
2
3 qt5_wrap_ui(UIS_HDRS ${UIS})
```

Finalement, la variable `UIS_HDRS` doit être passée en argument de la fonction `add_executable()` (lien 21) lors de la création du binaire.

### 1.6.d Utiliser des fichiers de ressources (.qrc)

De la même manière que les fichiers d'interface utilisateur, les fichiers de ressources doivent être traités avant de pouvoir être intégrés. La fonction est `qt5_add_resources` :

```
1 set(RESSOURCE hello-qt.qrc)
2
3 qt5_add_resources(RESSOURCE_HDR
  RESSOURCE)
```

Finalement, la variable `RESSOURCE_HDR` devra être passée en argument de la fonction `add_executable()` (lien 22) lors de la création du binaire.

### 1.6.e Création du binaire

La création du binaire se fait comme dans tout autre projet. Il ne faut simplement pas oublier les fichiers générés par le traitement des fichiers d'interface utilisateur (.ui) ou de ressources (.qrc) :

```
1 add_executable(hello-qt ${SRCS} ${HDRS}
  ${UIS_HDRS} ${RESSOURCE_HDR})
2 target_link_libraries(hello-qt Qt::
  Widgets Qt::OpenGL)
```

## 1.7 Conseils

### 1.7.a Version minimale

Il est conseillé d'indiquer une version minimale 3.X avec la fonction `cmake_minimum_required()`. Il peut encore être toléré de voir une version préalable 2.8, mais en règle générale, pour un nouveau projet, il vaut mieux utiliser la version la plus récente.



Il est surtout déconseillé d'indiquer les versions 2.4 et 2.6, qui, elles, sont vraiment trop anciennes.

#### 1.7.a.a Version minimale pour Qt

Il est fortement conseillé d'utiliser au minimum une version 2.8.12 lors d'une utilisation conjointe avec Qt. En effet, les fonctions ont été simplifiées rendant l'écriture du fichier `CMakeLists.txt` pour un projet Qt bien plus aisée.

#### 1.7.b Sous-projets

Lors de la création du `CMakeLists.txt` ne faites pas la supposition que votre projet sera toujours à la racine. Il est possible que le projet soit inclus à son tour dans un autre projet. Pour la même raison, ne modifiez pas les variables globales.

## 1.8 Empaquetage du projet

CMake, à travers CPack permet d'empaqueter le projet. L'empaquetage est le processus de réalisation d'un paquet (pour Linux), d'un installateur (pour Windows) ou simplement d'une archive.

Aucune installation supplémentaire n'est à faire. CPack est inclus à CMake.

### 1.8.a Configuration du projet pour CPack

Tout comme pour la compilation, nous allons utiliser le fichier `CMakeLists.txt` pour décrire à CPack comment empaqueter le projet.

Nous reprenons le deuxième projet et lui rajoutons la configuration nécessaire pour en produire un paquet (ou installateur pour Windows).

Le fichier `CMakeLists.txt` de la bibliothèque :

```
1 # Nous voulons un cmake "récent" pour
  # utiliser les dernières fonctionnalités
2 cmake_minimum_required(VERSION 3.0)
3
4 # Notre projet est étiqueté libhello
5 project(libhello)
```

```

6
7 # Crée des variables avec les fichiers à compiler
8 set(SRCS
9     hello.c
10    )
11
12 set(HEADERS
13     hello.h
14    )
15
16 add_library(hello ${SRCS} ${HEADERS})
17
18 # La bibliothèque peut être soit une bibliothèque statique, soit
19 # dynamique, donc on précise où installer pour les deux cas
20 install(TARGETS hello LIBRARY
21         DESTINATION lib
22         ARCHIVE
23         DESTINATION
24         lib)

```

Et le *CMakeLists.txt* pour le projet principal :

```

1 # Nous voulons un cmake "récent" pour utiliser les dernières fonctionnalités
2 cmake_minimum_required(VERSION 3.0)
3
4 # Notre projet est étiqueté hello
5 project(hello)
6
7 # On inclut notre bibliothèque dans le processus de CMake
8 add_subdirectory(libhello)
9
10 # Crée des variables avec les fichiers à compiler
11 set(SRCS
12     main.c
13    )
14
15 # Notre exécutable
16 add_executable(main ${SRCS})
17
18 # Et pour que l'exécutable fonctionne, il faut lui indiquer la bibliothèque

```

```

19 dont il dépend
20 target_link_libraries(main hello)
21
22 # Indique que l'on installe l'exécutable dans un dossier bin
23 install (TARGETS main DESTINATION bin)
24
25 # Section CPack
26 include (CPack)

```

Pour faire notre paquet, nous avons simplement indiqué comment installer notre programme avec la fonction `install()` : lien 23. Ensuite, pour pouvoir empaqueter le projet, nous avons invoqué CPack.

CPack utilise les appels à la fonction `install()` pour savoir quels sont les fichiers à inclure dans le paquet.



Pour obtenir un installateur sous Windows, il vous faudra installer le logiciel NSIS : lien 24.

Pour améliorer la configuration de votre installateur, vous pouvez utiliser de nombreuses variables décrites dans la documentation officielle : lien 25. Voici un court exemple d'utilisation :

```

1 SET(CPACK_PACKAGE_DESCRIPTION_SUMMARY "
2   CMake demo")
3 SET(CPACK_PACKAGE_VENDOR "LittleWhite")
4 SET(CPACK_PACKAGE_EXECUTABLES "Hello" "
5   Hello")

```



L'empaquetage n'est pas réalisé par CMake. Il faut exécuter une phase de compilation supplémentaire pour créer le package. Par exemple, si vous générez des Makefiles, cela peut se faire avec la commande `make package`.

Vous pouvez récupérer le projet ici : lien 26.

## 2 Conclusion

Nous avons vu comment configurer et compiler notre projet avec CMake afin de facilement produire les binaires, et ce, quels que soient le système ou les

outils de compilation utilisés. De plus, pour facilement redistribuer notre projet, nous avons ajouté la configuration de Cpack.

Retrouvez l'article d'**Alexandre Laurent** en ligne : lien 27



# Access

## Les derniers tutoriels et articles

# Un formulaire (état) qui contient de multiples fois le même sous-formulaire (sous-état), chacun avec une source différente

- Dans cette application, on a choisi de construire un formulaire qui contient 21 fois le même sous-formulaire avec pour chacun une source différente. Ces derniers sont construits à la volée.
- Un procédé identique sera présenté pour un état.
- On verra aussi comment il est possible, dans un formulaire continu, d'avoir une zone de liste modifiable dont le contenu est spécifique à chaque enregistrement.

## 1 Prérequis

Maîtriser la conception des formulaires et des états. (vos possibilités).

Être capable de « sentir le code VBA » : pas nécessairement savoir l'écrire, mais comprendre ce qui s'y passe (ce qui constitue une étape importante dans votre apprentissage de ce langage qui décuplera



Pour un problème de code dans un module, placez le curseur n'importe où dans l'instruction et pressez <F1>. L'aide Access s'ouvre alors à la bonne page.

## 2 Le but de l'exercice

On dispose d'une table qui contient les différents plats :

iPlatsPK	PlatNomF	PlatNomE	iCategoriesFK	Ligne
167	Bar salade	Salad bar	5	1
168	Buffet d'entrées	Starters	5	1
169	Bœuf à la crème	Cream beef	2	2
170	Bœuf à la tomate	Tomato beef	2	2
204	Aile de poulet pané	Breaded chicken wings	13	3
205	Blanc de poulet crème et champignons	Chicken breast with cream	13	3
206	Brechoth de blanc de poulet	Chicken breast	13	3

On voudrait alors construire les menus de la semaine pour les éléments de la carte des trois services journaliers :

À l'ouverture du formulaire : compléter la date

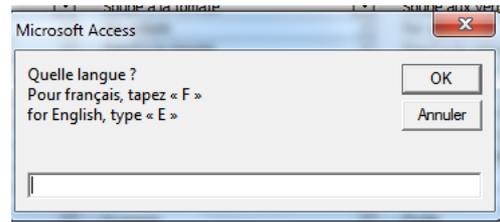
Date de départ :

Il vient

Lunch	Dinner	Midnight
Soupe	Soupe	Soupe
Entrée	Entrée	Entrée
Plat 1	Plat 1	Plat 1
Plat 2	Plat 2	Plat 2
Plat 3	Plat 3	Plat 3
Plat complet	Plat complet	Plat complet
Légumes 1	Légumes 1	Légumes 1
Légumes 2	Légumes 2	Légumes 2
Féculents	Féculents	Féculents
Garniture	Garniture	Garniture
Dessert	Dessert	Dessert

avec une liste déroulante limitée aux plats pertinents pour chaque cas, pour aboutir à ceci :

En un clic, produire un état synoptique des sept jours, au choix en français et en anglais :



En un clic, produire un état bilingue du menu d'un jour :

### 3 L'idée

Utiliser un formulaire « principal » qui contiendra 21 sous-formulaires (un pour chacune des trois

cartes de chacun des sept jours de la semaine) et récupérer ces données pour produire les états.

### 4 Les difficultés

1° Adapter la source de chaque sous-formulaire pour qu'il affiche les données de cette carte. Nous devons écrire quelques lignes de code VBA.

2° Adapter le contenu des zones de liste pour les limiter aux choix pertinents pour un item de ce type (soupe, entrée, plat...).

Le problème vient du fait que lorsqu'on modifie de manière dynamique le contenu de la zone de liste pour en restreindre les choix dans l'enregistrement actif, cette restriction entre automatiquement en vigueur pour tous les autres enregistrements du formulaire. Cela n'altère pas la valeur des choix antérieurs : l'item qui avait été choisi subsiste tel quel, mais si sa valeur ne figure pas dans la liste de celles ponctuellement permises, Access ne peut plus l'afficher. Au paragraphe 8, nous décrirons une astuce pour contourner le problème.

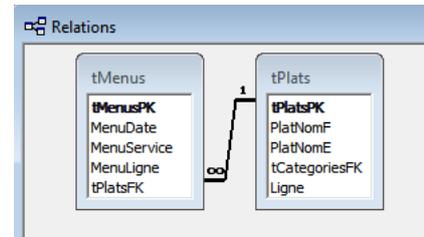
## 5 La table tMenu

Cette table pour stocker les données du plat choisi, pour chaque item des trois services de chaque jour (30 enregistrements par jour), par exemple pour le 8/10/2015 :

tMenuPK	MenuDate	MenuService	MenuLigne	tPlatsFK	NomF	NomE
231	8/10/15	1	1	168	Buffet d'entrées	Starters
232	8/10/15	1	2	170	Bœuf a la tomate	Tomato beef
233	8/10/15	1	3	204	Aile de poulet pan	Breaded chicken
234	8/10/15	1	4	264	Gambas à la plan	Grilled prawns
235	8/10/15	1	5	318	Lentilles au petit s	Lentil and pork
236	8/10/15	1	6	267	Aubergines à l'ail	Garlic eggplant
237	8/10/15	1	7	268	Aubergines grillée	Thym grilled eg
238	8/10/15	1	8	291	Gratin dauphinois	Potatoes gratin
239	8/10/15	1	9	332	Riz au jasmin	Jasmin Rice
240	8/10/15	1	10	333	Fromage	Cheese
241	8/10/15	2	1	167	Bar salade	Salad bar
242	8/10/15	2	2	169	Bœuf a la crème	Cream beef
243	8/10/15	2	3	204	Aile de poulet pan	Breaded chicken
244	8/10/15	2	4	253	Filet de thon grillé	Grilled tuna filet
245	8/10/15	2	5	309	Cassoulet Toulous	French cassoul
246	8/10/15	2	6	267	Aubergines à l'ail	Garlic eggplant
247	8/10/15	2	7	268	Aubergines grillée	Thym grilled eg
248	8/10/15	2	8	291	Gratin dauphinois	Potatoes gratin
249	8/10/15	2	9	332	Riz au jasmin	Jasmin Rice
250	8/10/15	2	10	333	Fromage	Cheese
251	8/10/15	3	1	167	Bar salade	Salad bar
252	8/10/15	3	2	169	Bœuf a la crème	Cream beef
253	8/10/15	3	3	205	Blanc de poulet cr	Chicken breast
254	8/10/15	3	4	222	Côte de porc char	Gherkins porc c
255	8/10/15	3	5	318	Lentilles au petit s	Lentil and pork
256	8/10/15	3	6	267	Aubergines à l'ail	Garlic eggplant
257	8/10/15	3	7	268	Aubergines grillée	Thym grilled eg
258	8/10/15	3	8	291	Gratin dauphinois	Potatoes gratin
259	8/10/15	3	9	332	Riz au jasmin	Jasmin Rice
260	8/10/15	3	10	333	Fromage	Cheese
261	5/10/15	1	1	168	Buffet d'entrées	Starters

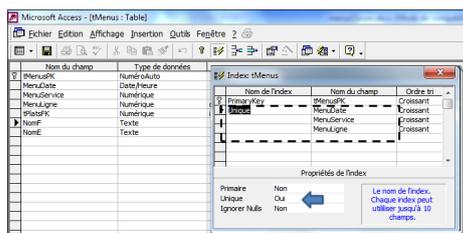


On pourrait considérer comme une redondance le fait que tMenu contienne les intitulés (français/anglais) puisque ceux-ci sont dans tPlats. On s'attendrait à une relation comme celle-ci pour obtenir le nom du plat :



Cependant, nous n'avons pas pu opter pour une telle solution, car dans un premier temps, lorsque les enregistrements sont créés, on ne connaît pas encore le plat qui sera choisi par l'utilisateur... et tPlatsFK vaut alors 0, par défaut.

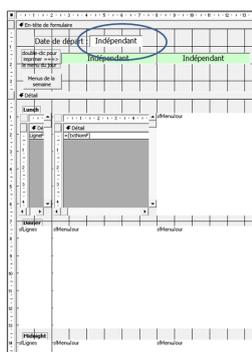
On évite les doublons comme ceci :



## 6 Le formulaire fMenuSemaine

### 6.1 Les pièces du puzzle

À l'ouverture, tous les contrôles sont invisibles sauf txtDateDepart.



Sa mise à jour fera passer la propriété Visible des autres contrôles à Oui.

La section Entête du formulaire contient sept contrôles avec une date qui correspondra à chaque jour de la semaine qui débute à la date de départ saisie par l'utilisateur.

Dans la section Détail, on trouve :

- 3 fois le sous-formulaire sfLignes destiné à afficher les 10 items d'une carte ;
- 21 fois le sous-formulaire sfMenuJour qui affichera le plat choisi pour chacun des 10 items des 21 cartes de la semaine (3 services fois 7 jours).

### 6.2 Le code déclenché par la mise à jour de txtDateDepart

```

1 Private Sub txtDateDepart_AfterUpdate()
2     Dim ctl As Control
3     Dim sSqlDebut As String
4     Dim sSql As String
5     Dim i As Integer
6     'Rendre les contrôles visibles
7     For Each ctl In Me.Controls
8         ctl.Visible = True
9     Next ctl
10    'Aménager les dates
11    DoCmd.SetWarnings False
12    For Each ctl In Me.Controls
13        If ctl.Name Like "txtJour#" Then
14            'Garnir les contrôles

```

```

15     ctl = Me.txtDateDepart + Right(
16         ctl.Name, 1)
17     'Créer (éventuellement) les
        enregistrements vierges avec
        ces dates dans la table
        tMenus
18     DoCmd.RunSQL "INSERT INTO tMenus
        ( MenuDate, MenuService,
        MenuLigne ) " -
        & "SELECT #" & Format(ctl,
        "mm/dd/yyyy") & "# AS
        Expr1, 1 AS Expr2,
        tMatrice.MenuLigne
        FROM tMatrice;"
19     DoCmd.RunSQL "INSERT INTO tMenus
        ( MenuDate, MenuService,
        MenuLigne ) " -
20         & "SELECT #" & Format(ctl,
        "mm/dd/yyyy") & "# AS
        Expr1, 2 AS Expr2,
        tMatrice.MenuLigne
        FROM tMatrice;"
21     DoCmd.RunSQL "INSERT INTO tMenus
        ( MenuDate, MenuService,
        MenuLigne ) " -
22         & "SELECT #" & Format(ctl,
        "mm/dd/yyyy") & "# AS
        Expr1, 3 AS Expr2,
        tMatrice.MenuLigne
        FROM tMatrice;"
23     End If
24 Next ctl
25 DoCmd.SetWarnings True
26 'Construire la source de chaque sous-
        formulaire et le contenu des
        listes
27 For Each ctl In Me.Controls
28     If ctl.Name Like "CTNR*" Then
29         'source
30         ctl.Form.RecordSource = "SELECT
        tMenus.* FROM tMenus " -
31             & "WHERE MenuDate
        =[Formulaires
        ]![
        fMenuSemaine
        ]![txtJour" &
        Mid(ctl.Name
        , 5, 1) & "]"
        -
32             & "AND
        MenuService
        =" &
        Right(
        ctl.Name
        , 1) & "
        ORDER
        BY
        tMenus.
        MenuLigne
        ;"
33     'Liste
34     ctl.Form!cboPlat.RowSource = "
        SELECT tPlatsPK, PlatNomF,
        PlatNomE FROM tPlats " -
35         & "WHERE Ligne=Replace([
        Forms]![fMenuSemaine]!"
        & ctl.Name & ".[form]![
        txtMenuLigne],6,7);"
36
37     End If
38 Next ctl
39 End Sub

```

### Explication du code

**7-9** : on lance une boucle sur tous les contrôles du formulaire pour les rendre visibles.

**10-25** : modification des propriétés des contrôles en fonction des dates de la semaine.

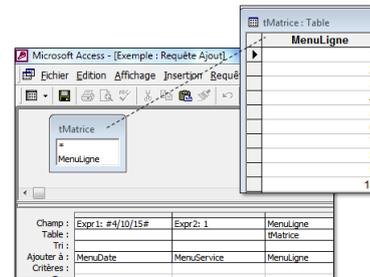
**15** : les contrôles nommés txtDate<sub>i</sub> où « i » varie de 0 à 6



Pour calculer la date qui convient, il suffit d'ajouter à la date *txtDateDepart* un nombre de jours égal au chiffre qui termine le nom du contrôle.

**16-22** : ici, pour chacune des sept dates, on va compléter (éventuellement) la table *tMenus* pour qu'elle contienne **3** (les trois services) x **10** (les dix items de la carte) soit **30** enregistrements pour chaque journée.

Les requêtes construites à la volée sont du type de celle-ci par exemple pour le lunch (code service = 1) du 4/10/2015

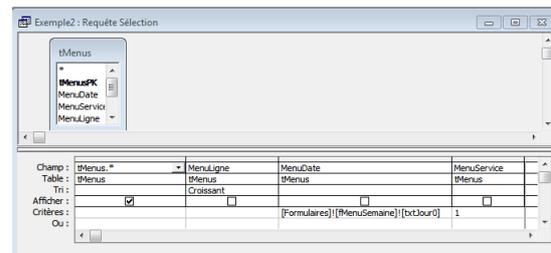


De deux choses l'une :

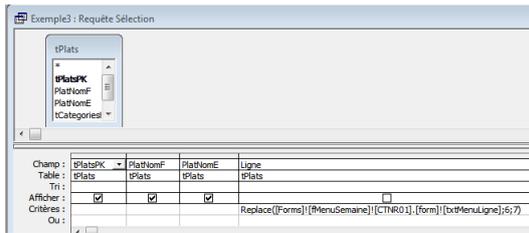
- les enregistrements existent déjà et l'ajout sera rejeté (voir définition de *tMenus*);
- des enregistrements seront ajoutés pour les dix items de la carte de ce service de cette date, prêts à accueillir le plat qui sera choisi par l'utilisateur. (À ce stade, *tPlatsFk* vaut 0.)

**27-38** : adaptation des 21 sous-formulaires.

**30-32** : on crée la source de chacun. En l'occurrence, la portion de *tMenus* qui le concerne. Par exemple pour le lunch du jour0 (CTNR01, où « 0 » indique le jour et « 1 » le lunch), cette requête créée à la volée :



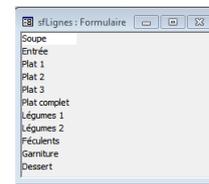
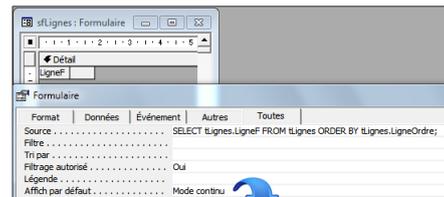
**34-35** : dans la foulée, on aménage le contenu des zones de liste. Voici un exemple des 21 requêtes construites à la volée, il se rapporte à la liste des items pour le lunch du jour0 :



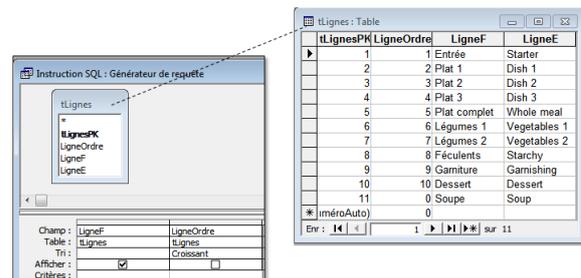
N.B. Les lignes 6 et 7 de la carte concernent les légumes. Dans *tPlats* ceux-ci ont la valeur 7 en colonne *Ligne*.

## 7 Le sous-formulaire sfLignes

Il est destiné à afficher les items d'une carte dans le formulaire principal :



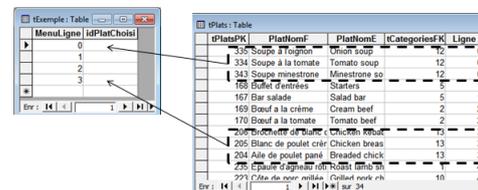
Sa source :



Sa construction :

## 8 La problématique des zones de liste à contenu variable dans un formulaire continu

Le problème vient du fait que lorsqu'on modifie de manière dynamique le contenu de la zone de liste pour en restreindre les choix dans l'enregistrement actif, cette restriction entre automatiquement en vigueur pour tous les autres enregistrements du formulaire. Cela n'altère pas la valeur des choix qui ont été faits dans les autres enregistrements : l'item qui avait été choisi subsiste tel quel, mais si sa valeur ne figure pas dans la liste de celles ponctuellement permises, Access ne peut plus l'afficher.

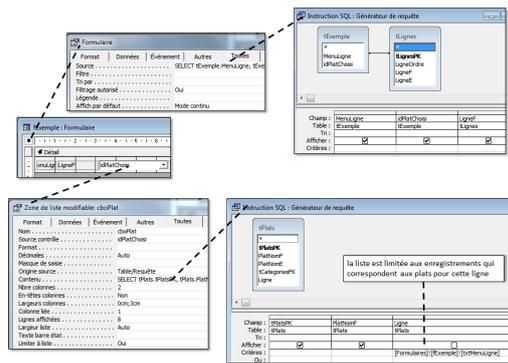


on veut garnir la colonne *idPlatChoisi* en picorant dans une liste dont le choix serait limité aux plats qui correspondent à la ligne indiquée.

### 8.1 Un exemple pour bien se comprendre

Supposons que dans cette table :

On pense d'abord à un formulaire comme celui :



le choix s'affiche à nouveau... et les autres disparaissent !

## 8.2 Que se passe-t-il ?

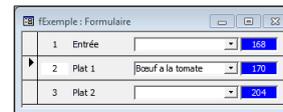
Pour bien comprendre, nous allons ajouter un contrôle au formulaire pour afficher la valeur contenue dans *idPlatChoisi*.

Voici :

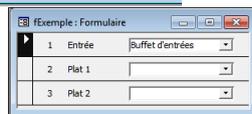
Avec ce code sur l'événement « Sur activation » pour actualiser le contenu de la liste à chaque enregistrement :

```
1 Option Compare Database
2 Option Explicit
3
4 Private Sub Form_Current()
5     Me.cboPlat.Requery
6 End Sub
```

Ce qui donnerait ceci :

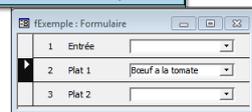


Quand on est positionné sur la ligne 2, la requête qui fixe le contenu de la zone de liste limite le choix à 169 et 170 :



tPlatsPK	PlatNomF	PlatNomE	CategoriesFK	Ligne
335	Soupe à l'oignon	Onion soup	12	0
334	Soupe à la tomate	Tomato soup	12	0
343	Soupe minestrone	Minestrone so	12	0
168	Buffet d'entrées	Starters	5	1
169	Bœuf à la crème	Cream beef	5	1
170	Bœuf à la tomate	Tomato beef	2	2
206	Brochette de blanc	Chicken kebab	13	3
205	Blanc de poulet cré	Chicken breas	13	3
204	Aile de poulet pané	Breaded chick	13	3
235	Epaule d'agneau rôti	Roast lamb sh	1	4
223	Steak de porc grillé	Grilled pork ch	10	4

MAIS pour la ligne suivante :



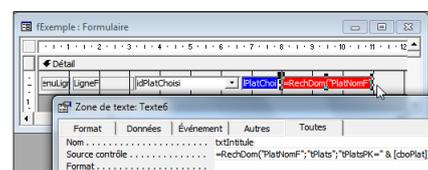
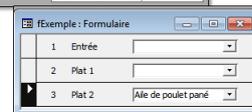
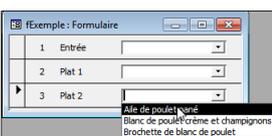
... et ni 168 ni 204 ne trouvent leur intitulé !

## 8.3 Une astuce pour sortir de l'impasse

Nous allons ajouter un nouveau contrôle au formulaire :

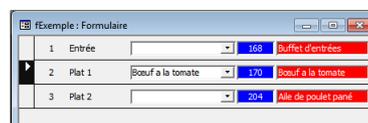
la liste est bien limitée aux bons choix, mais le choix qui avait été opéré pour « Entrée » n'est plus affiché !

De même :



avec comme propriété *Source contrôlée* une fonction de domaine qui recherche dans *tPlats* l'intitulé correspondant à la valeur mémorisée dans *cboPlat* (soit *idPlatChoisi* de cet enregistrement).

Et voilà le travail :

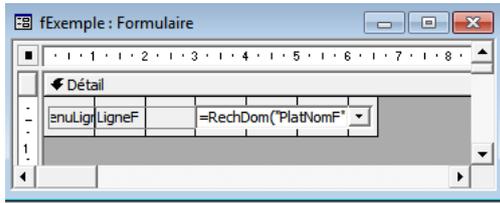


Bref, seul l'enregistrement actif laisse apparaître le choix opéré.

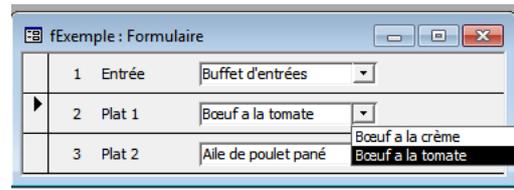
D'ailleurs, si on revient sur l'enregistrement 2 :

### 8.4 Une dernière retouche pour faire « joli »

On réduit la largeur de la zone de liste pour ne laisser apparaître que son carré  que l'on place à droite de la zone de texte qui contient l'intitulé :



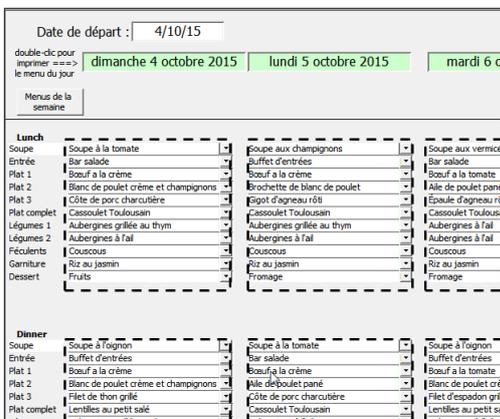
Et on obtient finalement une « illusion » de zone de liste modifiable :



Pas beau ça ?

## 9 Le sous-formulaire sfMenuJour

Il est destiné à sélectionner le plat de chaque item des 21 cartes :



de la carte (uniquement les soupes sur la 1<sup>re</sup> ligne, uniquement des entrées sur la 2<sup>e</sup>, etc.)

tPlatsPK	PlatNomF	PlatNomE	tCategoriesFK	Ligne
335	Soupe à l'oignon	Onion soup	12	0
334	Soupe à la tomate	Tomato soup	12	0
343	Soupe minestrone	Minestrone so	12	0
168	Buffet d'entrées	Starters	5	1
167	Bar salade	Salad bar	5	1
169	Boeuf à la crème	Cream beef	2	2
170	Boeuf à la tomate	Tomato beef	2	2
206	Brochette de blanc c	Chicken kebab	13	3
205	Blanc de poulet crêr	Chicken breas	13	3
204	Aile de poulet pané	Breaded chick	13	3
235	Épaule d'agneau rôti	Roast lamb sh	1	4
223	Côte de porc grillée	Grilled pork ch	10	4

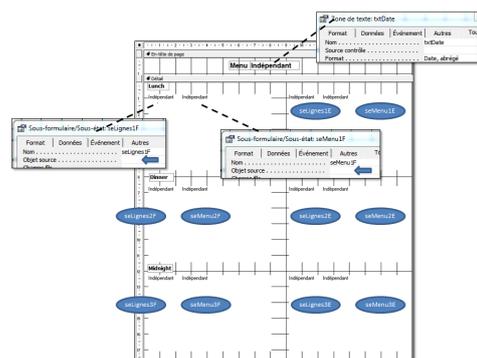
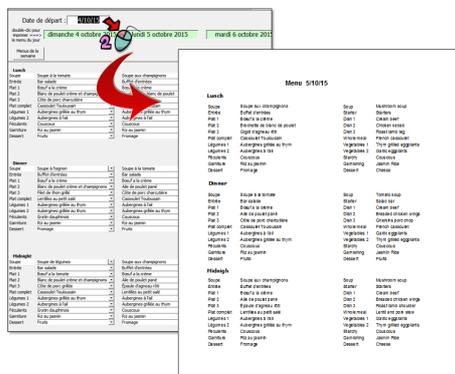
Sur chaque ligne, une zone de liste modifiable qui permet de choisir un plat compatible avec l'item

Pour ce qui est de la zone de liste modifiable, ce sous-formulaire est conçu suivant la technique décrite au chapitre précédent.

## 10 L'état eMenuJour

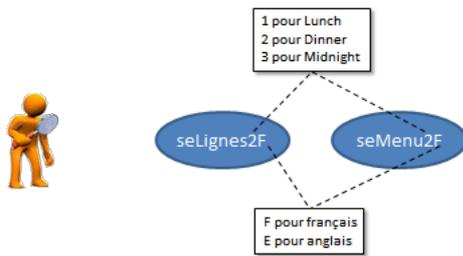
L'idée, c'est d'afficher le formulaire *fMenu* et d'obtenir, en double-cliquant sur une date, un état qui donne la carte bilingue de cette date :

### 10.1 Un squelette : eMenuJourVierge



L'objectif est de récupérer les données déjà disponibles dans le formulaire.

Remarquez comment sont nommés les conteneurs de sous-état



L'objet source du conteneur n'est pas complété.

## 10.2

Pour confectionner l'état, on va prendre une copie de ce squelette, nous l'appellerons *eMenuJour* et nous compléterons à la volée la date dans *txtDate* et l'objet source dans chacun des douze conteneurs.

La structure adoptée pour nommer les contrôles nous permettra de factoriser le code.

## 10.3 Le code du double-clic sur une date de fMenuSemaine

```

1 Option Compare Database
2 Option Explicit
3
4 Private Sub txtJour0_DblClick(Cancel As Integer)
5     Me.txtDateDepart.SetFocus
6     Call MenuJour(0)
7 End Sub
8
9 Private Sub txtJour1_DblClick(Cancel As Integer)
10    Me.txtDateDepart.SetFocus
11    Call MenuJour(1)
12 End Sub
13
14 Private Sub txtJour2_DblClick(Cancel As Integer)
15    Me.txtDateDepart.SetFocus
16    Call MenuJour(2)
17 End Sub
18
19 Private Sub txtJour3_DblClick(Cancel As Integer)
20    Me.txtDateDepart.SetFocus
21    Call MenuJour(3)
22 End Sub
23 Private Sub txtJour4_DblClick(Cancel As Integer)
24    Me.txtDateDepart.SetFocus
25    Call MenuJour(4)
26 End Sub
27
28 Private Sub txtJour5_DblClick(Cancel As Integer)
29    Me.txtDateDepart.SetFocus
30    Call MenuJour(5)
31 End Sub
32

```

```

33 Private Sub txtJour6_DblClick(Cancel As Integer)
34     Me.txtDateDepart.SetFocus
35     Call MenuJour(6)
36 End Sub
37
38 Public Sub MenuJour(NumJour As Integer)
39
40     Dim ctl As Control
41     Dim i As Integer
42
43     'Geler temporairement l'affichage à l'
44     'écran et les messages
45     Application.Echo False
46     DoCmd.SetWarnings False
47
48     'Initialiser eMenuJour
49     DoCmd.CopyObject , "eMenuJour",
50     acReport, "eMenuJourVierge"
51
52     'Ouvrir eMenuJour en mode construction
53     DoCmd.OpenReport "eMenuJour",
54     acViewDesign
55
56     'Dupliquer seLignes en seLignes_i_F et
57     'E
58     -----
59     --
60     DoCmd.OpenReport "seLignes",
61     acViewDesign
62     'En français
63     Reports!seLignes!txtLigne.
64     ControlSource = "LigneF"
65     DoCmd.Save acReport, "seLignes"
66     For i = 1 To 3
67         'Créer les sous-états
68         DoCmd.CopyObject , "seLignes" & i &
69         "F", acReport, "seLignes"
70         'Aménager les ObjectSource des sous-
71         états seLignes_i
72         Reports!eMenujour("seLignes" & i & "
73         F").SourceObject = "État.
74         seLignes" & i & "F"
75     Next i
76     'En anglais
77     Reports!seLignes!txtLigne.
78     ControlSource = "LigneE"
79     DoCmd.Save acReport, "seLignes"
80     For i = 1 To 3
81         'Créer les sous-états
82         DoCmd.CopyObject , "seLignes" & i &
83         "E", acReport, "seLignes"
84         'Aménager les ObjectSource des sous-
85         états seLignes_i
86         Reports!eMenujour("seLignes" & i & "
87         E").SourceObject = "État.
88         seLignes" & i & "E"
89     Next i
90     DoCmd.Close acReport, "seLignes"
91
92     'Dupliquer seMenu
93     -----
94     DoCmd.OpenReport "seMenu",
95     acViewDesign
96     'En français
97     Reports!seMenu!txtNom.ControlSource =
98     "NomF"
99     Reports!seMenu!txtNom.FontSize = 8
100    DoCmd.Save acReport, "seMenu"
101    For i = 1 To 3
102        'Créer les sous-états
103        DoCmd.CopyObject , "seMenu" & i & "F
104        ", acReport, "seMenu"

```

```

86      'Aménager les ObjectSource des sous-
      états seMenu_i
87      Reports!eMenujour("seMenu" & i & "F"
      ).SourceObject = "État.seMenu" &
      i & "F"
88      Next i
89      'En anglais
90      Reports!seMenu!txtNom.ControlSource =
      "NomE"
91      DoCmd.Save acReport, "seMenu"
92      For i = 1 To 3
93          'Créer les sous-états
94          DoCmd.CopyObject, "seMenu" & i & "E"
          , acReport, "seMenu"
95          'Aménager les ObjectSource des sous-
          états seMenu_i
96          Reports!eMenujour("seMenu" & i & "E"
          ).SourceObject = "État.seMenu"
          & i & "E"
97      Next i
98      DoCmd.Close acReport, "seMenu"
99
100     'Aménager les dates
101     Reports!eMenujour!txtDate.
      ControlSource = _
102         "-" & Format(Me("
          txtJour" &
          NumJour), "mm/
          dd/yyyy") & "-"
103
104     'Sauvegarder et rouvrir en
      construction
105     DoCmd.Save acReport, "eMenujour"
106     DoCmd.Close acReport, "eMenujour"
107     DoCmd.OpenReport "eMenujour",
      acViewDesign
108
109     'Aménager les sources
110     For Each ctl In Reports!eMenujour.
      Controls
111         If ctl.Name Like "seMenu*" Then
112             Reports!eMenujour!(ctl.Name).
      Report.RecordSource = _
113                 Me("CTNR" & NumJour
          & Mid(ctl.Name
          , 7, 1)).Form.
          RecordSource
114         End If
115     Next ctl
116
117     'Rétablir l'affichage et les messages
118     Application.Echo True
119     DoCmd.SetWarnings True
120
121     'Afficher l'état
122     DoCmd.OpenReport "eMenuJour ",
      acViewPreview
123 End Sub

```

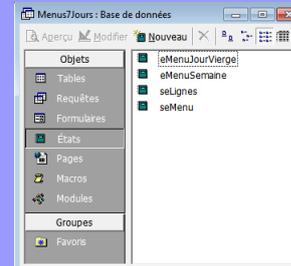
#### Complément aux commentaires du code

**4 - 33** : le double-clic sur chacune des sept dates appelle la sous-routine *MenuJour* en donnant son numéro d'ordre comme paramètre (0 pour le 1<sup>er</sup> jour,

6 pour le dernier).

**35 - fin** : la sous-routine *Menujour*.

À l'ouverture, la base contient uniquement ces états qui vont servir de modèle :



Dans le code qui suit, nous allons construire à la volée toute une série d'états... que nous supprimerons à la fermeture.

**50 - 54** : on crée *eMenuJour* à l'image de *eMenuJourVierge*.

**56 - 78** : on garnit les conteneurs de type

seLignes\_i\_langue

d'abord en français (60 - 66), puis en anglais (70 - 78).

**82 - 104** : on suit la même logique pour garnir les conteneurs de type

seMenu\_i\_langue

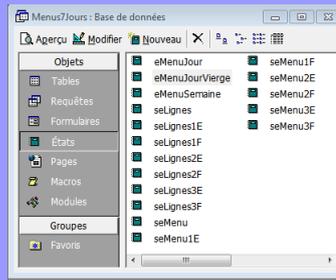
**111 - 113** : dans certaines circonstances (et c'est le cas ici) Access a du mal à s'y retrouver lorsqu'on fait référence à des contrôles d'un objet que l'on est en train de construire à la volée (en l'occurrence *eMenujour*). Pour cette raison, on sauvegarde, on ferme et on rouvre en construction *eMenujour* avant de compléter sa construction.

**117 - 122** : on s'occupe ici de l'objet source de chacun des six conteneurs

seMenu\_i\_langue

En fait, on récupère pour chacun, l'objet source du conteneur correspondant sur le formulaire *fMenuSemaine*.

Après l'exécution de ce code, la fenêtre des objets « États » a évolué :



C'est la fermeture du formulaire *fMenuSemaine* qui provoquera le « nettoyage » de la base

```

1 Private Sub Form_Close()
2     Dim db As DAO.Database, doc As DAO.Document
3     'Fermer eMenuJour s'il est encore ouvert
4     DoCmd.Close acReport, "eMenuJour"
5     Set db = CurrentDb
6     For Each doc In db.Containers("Reports").Documents
7         If doc.Name Like "seMenu#*" Or doc.Name Like "seLignes#*" Or doc.Name
8             = "eMenuJour" Then
9             DoCmd.DeleteObject acReport, doc.Name
10        End If
11    Next doc
12    Set db = Nothing
13    DoCmd.Restore
14 End Sub

```

## 11 L'état eMenuSemaine



On complète les contrôles variables à la volée, la technique est la même que pour eMenuJour.

### 11.1 Le code du clic sur le bouton Menu de la semaine

```

1 Private Sub btSemaine_Click()
2     Dim i As Integer
3     Dim j As Integer
4     Dim Reponse As String
5     'Demander dans quelle langue
6     Reponse = UCase(InputBox("Quelle
7         langue ?" & Chr(10) & Chr(13) _
8         & "Pour français, tapez
9         < F >" &
10        Chr(10) &
11        Chr(13) _
12        & "for English,
13        type < E >"
14        ))
15     If Reponse <> "F" And Reponse <> "E"
16     Then
17         MsgBox "Choix de langue invalide " &
18         Chr(10) & Chr(13) _
19         & "Procédure interrompue."

```

```

12     Exit Sub
13 End Sub
14 Application.Echo False
15 'Aménager les contrôles variables de
16     eMenuSemaine
17 'Créer les sous-états selon la langue
18 'Formater seLignes en fonction
19     de la langue
20 DoCmd.SetWarnings False
21 DoCmd.OpenReport "seLignes", acDesign
22 Reports!seLignes!txtLigne.
23     ControlSource = "Ligne" & Reponse
24 DoCmd.Save acReport, "seLignes"
25 DoCmd.Close acReport, "seLignes"
26 DoCmd.SetWarnings True
27 'Formater seMenu en fonction de
28     la langue
29 DoCmd.SetWarnings False
30 DoCmd.OpenReport "seMenu", acDesign
31 Reports!seMenu!txtNom.ControlSource =
32     "Nom" & Reponse
33 Reports!seMenu!txtNom.FontSize = 6
34 DoCmd.Save acReport, "seMenu"
35 DoCmd.Close acReport, "seMenu"
36 DoEvents
37 'Démultiplier
38 For i = 1 To 3
39     For j = 0 To 6
40         DoCmd.CopyObject, "seMenu" & i &
41         j, acReport, "seMenu"
42     Next j
43 Next i
44 DoCmd.SetWarnings True
45 'Aménager les contrôles de dates
46 DoCmd.OpenReport "eMenuSemaine",
47     acViewDesign

```

```

41 Reports!eMenuSemaine!txtDu.
   ControlSource = "=" & Format(Me.
42   txtDateDepart, "mm/dd/yyyy") & "#"
Reports!eMenuSemaine!txtAu.
   ControlSource = "=" & Format(Me.
43   txtDateDepart + 6, "mm/dd/yyyy") &
   "#"
44 For i = 0 To 6
   Reports!eMenuSemaine("txtDate" & i).
   ControlSource = "=" & Format(Me
45   .txtDateDepart + i, "mm/dd/yyyy
   ") & "#"
46 Next i
47 'Aménager les sources
48 For i = 0 To 6
   Reports!eMenuSemaine("ctnrLunch" & i
49   ).Report.RecordSource = Me("CTNR
   " & i & 1).Form.RecordSource
Reports!eMenuSemaine("ctnrDinner" &
50   i).Report.RecordSource = Me("
   CTNR" & i & 2).Form.RecordSource
Reports!eMenuSemaine("ctnrMidnight"
51   & i).Report.RecordSource = Me("
   CTNR" & i & 3).Form.RecordSource
52 Next i
53 DoCmd.Save acReport, "eMenuSemaine"
54 'Afficher
55 Application.Echo True
56 DoCmd.OpenReport "eMenuSemaine",
   acViewPreview
End Sub

```

## 12 Téléchargement

La db (format Access2000) qui a servi à mettre ce processus au point peut être téléchargée ici : [lien 28](#).  
 Retrouvez l'article de **Claude Leloup** en ligne : [lien 29](#)



# Delphi

## Les derniers tutoriels et articles

# Le développement Intraweb avec Delphi

Delphi est essentiellement connu comme étant un environnement de développement permettant de construire rapidement des applications riches et robustes pour Windows, Mac et mobiles. Son utilisation pour la réalisation d'applications destinées à s'exécuter dans un navigateur est moins connue. Il dispose pourtant de quelques *frameworks* lui permettant d'apporter ses atouts également à ce type de projet. Je propose dans cet article de présenter **Intraweb**, qui est un produit de la société Atozed software (lien 30) et qui est probablement le *framework* le plus connu puisque sa version *bundle* est intégrée à Delphi.

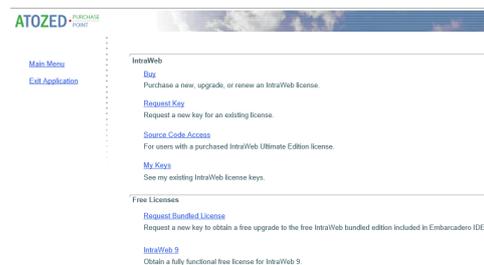
## 1 Présentation

**Intraweb** est un *framework* permettant de créer des applications HTML avec Delphi. L'idée était de porter les points forts de Delphi également pour le développement web. Avec ce *framework*, nous allons pouvoir écrire simplement des applications web et les déployer, soit :

- avec un serveur web inclus ;
- sous forme de dll ISAPI pouvant être déployée sous IIS ;
- en ASPx (pour la dernière version **d'Intraweb**).

## 2 d'Intraweb

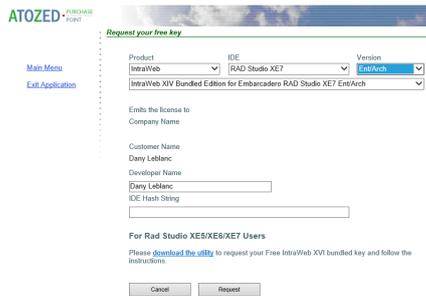
**Intraweb** fait partie du package standard de RAD Studio. Cependant, il est préférable de ne pas garder cette version, mais d'aller télécharger gratuitement la dernière version sur le site de l'éditeur, à savoir Atozed Software (lien 31). Ceci permettra de s'enregistrer et d'être informé des mises à jour du produit, voire de récupérer celles disponibles pour la version *bundle*. Je vous propose dans ce chapitre de détailler la procédure à suivre pour faire cette mise à jour.



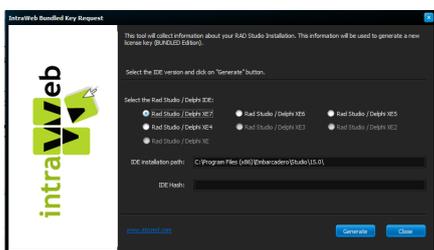
La première chose à faire est de demander notre clé de licence. Pour cela, nous allons sur la partie du site Atozed purchase point (lien 32). La première chose à renseigner est notre adresse de courriel. Après avoir cliqué sur « Next », soit notre adresse est déjà connue et nous pouvons saisir notre mot de passe, soit le site propose de poursuivre pour nous enregistrer. Je passe la procédure classique de création d'un compte utilisateur. Donc, une fois le mot de passe saisi, on arrive sur la page récapitulant nos informations. Tout en bas de cette page, il existe un lien nommé **Intraweb**, cliquons dessus. Nous arrivons alors sur cette page :

La partie haute permet d'acheter une licence payante et de gérer nos éventuelles licences déjà acquises. Nous allons nous intéresser à la partie du bas et cliquer sur « Request Bundled License » (on peut noter au passage qu'Atozed propose une ancienne version complète et gratuite).

Dans la page suivante, on va devoir renseigner les informations concernant notre licence RAD Studio et surtout leur indiquer la chaîne « IDE hash string » (eh oui ce téléchargement n'est accessible que pour les détenteurs d'une licence RAD Studio et en plus il sera différent suivant que nous avons la version Professionnelle ou Entreprise).



Pour obtenir la chaîne de caractères à entrer dans cette page, Atozed propose un petit outil en téléchargement (nous le téléchargeons même si nous avons déjà fait la procédure pour une précédente version de Delphi). Cet outil propose une interface toute simple :



Nous choisissons notre version de RAD Studio, nous pouvons modifier le répertoire d'installation si ce n'est pas celui qui est proposé, et nous cliquons sur « Generate ». Voilà, la clé est générée, et l'outil a même été jusqu'à la copier dans le presse-papier ; nous n'avons donc plus qu'à la coller sur la page web précédente. Enfin, nous cliquons sur « Request ». Nous allons recevoir la clé d'enregistrement dans notre boîte mail. Elle restera également disponible sur le lien « My keys » de l'écran principal d'Atozed purchase point.

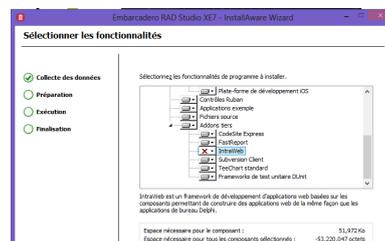
Ceci étant fait, nous pouvons nous rendre sur la page des téléchargements pour récupérer la dernière version disponible pour notre édition de RAD Studio.

L'étape suivante va consister à supprimer la version qui a été installée avec Delphi (ceci sera d'ailleurs la procédure à suivre pour toute mise à

### 3 Première application

Pour commencer notre application, créons le projet **Intraweb**. Pour ce faire, comme pour n'importe quel projet, cela commence par la boîte de dialogue de nouveau projet, dans laquelle nous pouvons trouver une ligne « Intraweb » que nous sélectionnons :

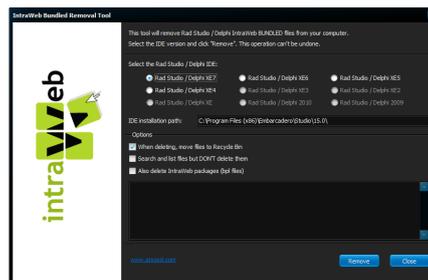
jour d'une version d'**Intraweb**). Il faut donc pour cela retourner dans l'installateur de RAD Studio, de choisir « Modifier » et de désélectionner « Intraweb » :



Nous validons ensuite l'installation et attendons la fin de cette dernière pour continuer.

Mais ceci ne sera pas suffisant, nous allons devoir supprimer les fichiers laissés sur le disque par le désinstallateur. Heureusement, depuis quelque temps Atozed fournit un petit outil qui se charge de cette tâche fastidieuse et délicate. Cet outil est téléchargeable ici : RemovalTool (lien 33).

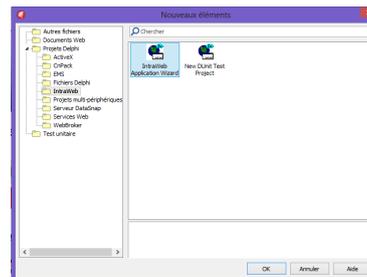
L'utilisation de cet outil est encore une fois très simple :



On choisit notre version de Delphi, puis on clique sur le bouton « Remove ». La liste des fichiers supprimés défile. Une fois terminé, on ferme cet outil.

Il ne reste donc plus qu'à lancer l'installateur d'**Intraweb** que nous avons précédemment téléchargé. Durant cette installation, la clé reçue par mail sera demandée pour pouvoir continuer.

Voilà, nous sommes prêts à développer des applications web avec Delphi et **Intraweb**.



Sélectionner « Intraweb Application Wizard », un nouveau dialogue s'affiche :



Nous pouvons ici sélectionner le type d'application que nous souhaitons réaliser :

- Standalone Application : il s'agit d'une application incluant un serveur web et qui peut donc s'exécuter sans passer par IIS. Ce type d'application est très pratique pour déboguer notre programme. Mais ce type d'application peut très bien être déployé en production, puisqu'il y a possibilité de l'installer en tant que service Windows. Ceci peut s'avérer une bonne solution d'ailleurs pour éviter de monter tout un serveur IIS si on n'a pas d'autre besoin de publication que notre application ;
- ISAPI Extension : ce type d'application va permettre de créer une dll qui sera ensuite déployée sur un serveur IIS. Ceci est pratique si vous avez déjà un serveur IIS, et indispensable si vous devez diffuser plusieurs applications sur le même port d'écoute.

Nous avons également la possibilité de choisir des options :

- Pool Data Connection : si cette option est cochée, le template de projet va générer les méthodes nécessaires permettant de gérer un pool de Datamodules qui pourront être attribués à la volée lorsque le programme en aura besoin. Nous verrons un peu plus bas comment gérer cela ;
- Use ISAPIThreadPool : option disponible uniquement dans le cas d'un projet de type ISAPI Extension, elle ajoute juste un uses permettant d'utiliser la fonctionnalité IIS qui permet de threader les pools.

Nous choisissons pour l'exemple une application de type *standalone* et nous ne nous occupons pas des options pour le moment. Après avoir cliqué sur « Ok », notre nouveau projet est créé.

Le *template* de création du projet nous a donc créé trois unités que nous verrons en détail plus bas :

- ServerController, qui représente le serveur ;
- Unit1, qui est notre formulaire principal ; nous le renommons *ufrmMain* et le formulaire s'appellera *frmMain* ;
- UserSessionUnit, qui est un *datamodule* qui va pouvoir contenir tous les éléments nécessaires à la session de l'utilisateur.

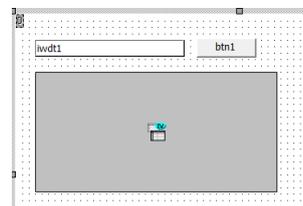
Habituellement lorsqu'on veut faire une première démo sur une technologie web, on fait un classique « Hello world ». Avec Delphi et le *framework IntraWeb*, je vais me permettre une démo un tout petit peu plus ambitieuse. Je vous propose donc de faire une petite application toute simple d'affichage d'une table d'une base de données avec possibilité de recherche.

Nous allons donc dans un premier temps nous intéresser uniquement à l'unité « *ufrmMain* » qui va être notre formulaire principal. Comme dans un projet VCL classique, nous disposons d'un *dfm* sur lequel nous allons pouvoir poser nos composants. La première petite différence va se situer sur la palette de composants à utiliser. Il ne va pas falloir bien entendu utiliser les composants de la VCL, mais leurs équivalents qui se trouvent dans les palettes dont le nom commence par « *Intraweb* ».



 Les noms des composants IntraWeb commencent par « *TIW* ».

Nous posons donc sur notre formulaire un *TIWEdit*, un *TIWButon* et un *TIWDbgrid*, pour obtenir quelque chose comme ceci :



Nous nommerons ces composants respectivement *iwdt1*, *btn1* et *grd1*.

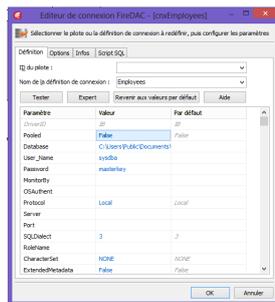
Pour les données, nous allons utiliser la base exemple *Employee.gdb*, que nous accéderons avec les composants *Firedac*, mais bien entendu cela reste valable avec toute autre base de données et également avec les composants *dbExpress*, ou autres.

Il va donc falloir ajouter quelques composants, non visuels ceux-ci, à notre formulaire :

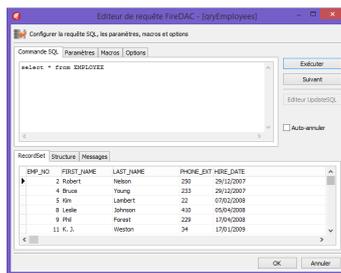
Tout d'abord pour l'accès à notre table :

- un *TFDConnection* que nous nommons *cnxEmployees* ;
- un *TFDGuidxWaitCursor* que nous nommons *fdw1* ;
- un *TFDQuery* que nous nommons *qryEmployees*.

Un clic droit sur le *TFDConnection* permet d'accéder à l'éditeur de connexion qui va permettre de paramétrer la connexion à la base de données exemple :



Une fois ceci fait, on va sur le composant TFD-Query pour saisir notre requête. Un double-clic sur ce dernier permet de saisir et de tester celle-ci :



Voilà, c'est tout pour l'accès à la table.

Une fois ce dialogue validé, on peut passer la propriété « Active » du composant à « True ».

Posons un autre composant qui va permettre de lier les composants visuels à ces données :

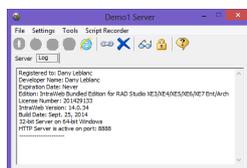
— un Tdatasource.

Le paramétrage de ce composant est également très simple puisqu'il suffit de renseigner la valeur de la propriété « Dataset » du composant TDataSource par le nom du composant TFDQuery (qryEmployees dans notre exemple).

De plus, la propriété possède une liste déroulante permettant de choisir le nom.

Pour terminer, renseignons de la même manière la propriété « Datasource » du TIWDBGrid avec le nom du TDataSource. Toujours sur le TIWDBGrid, on va mettre la propriété « UseFrame » à false.

Voilà, un petit appui sur la touche de fonction F9 permet de compiler et exécuter notre application, ce qui va nous donner ceci :



## 4 Synoptique d'un projet Intraweb

À présent, arrêtons-nous un peu sur ce qui a été généré par le template de projet. Nous venons de voir l'unité UfrmMain.pas, qui est le formulaire principal par défaut de notre application, mais suivant les options choisies deux ou trois autres unités ont

Rien à voir avec notre formulaire bien entendu, il s'agit en fait du serveur intégré à **Intraweb** (rappelons-nous : nous avons choisi une application *standalone*). Il faut faire à nouveau F9 pour voir le navigateur s'ouvrir et afficher le résultat qui devrait ressembler à cela :

EMP_ID	FIRST_NAME	LAST_NAME	PHONE_EXT	HIRE_DATE	JOB	DEPARTMENT	COUNTRY	SALARY	FULL_NAME
1	Robert	Debra	520	29-12-2007	REP	USA	USA	10500	Robert Debra
2	James	Clara	521	29-12-2007	REP	USA	USA	9500	James Clara
3	Kim	Lambert	522	07-02-2008	REP	USA	USA	10270	Kim Lambert
4	Young	Michael	430	07-04-2008	REP	USA	USA	8440	Young Michael
5	Paul	Franz	523	17-04-2008	REP	USA	USA	15000	Paul Franz
6	Kerr	Thomas	440	07-07-2008	REP	USA	USA	9000	Kerr Thomas
7	Ferris	Lee	516	01-01-2009	REP	USA	USA	9790	Ferris Lee
8	Stewart	Mark	527	04-06-2009	REP	USA	USA	8442.03	Stewart Mark
9	Kolton	Tracy	511	11-09-2009	REP	USA	USA	2152.03	Kolton Tracy
10	Chris	Paguspoulos	587	01-01-2009	REP	USA	USA	8951	Chris Paguspoulos
11	Pat	Haas	488	11-06-2009	REP	USA	USA	1110.18	Pat Haas
12	Ann	Benson	5	01-02-2010	REP	USA	USA	2397	Ann Benson
13	Paul	De Souza	44	01-02-2010	REP	USA	USA	8942.03	Paul De Souza
14	Paul	Baerlin	5	21-03-2010	REP	USA	USA	8157.1	Paul Baerlin
15	Roger	Swamy	6	21-04-2010	REP	USA	USA	1020.03	Roger Swamy
16	Chloe	Roemer	64	01-07-2010	REP	USA	USA	9211.08	Chloe Roemer
17	Julia	Phung	316	01-06-2010	REP	USA	USA	26514.38	Julia Phung
18	Mark	Rosenbaum	339	01-08-2010	REP	USA	USA	3049.9	Mark Rosenbaum
19	Chloe	Rosenbaum	310	09-08-2010	REP	USA	USA	10400	Chloe Rosenbaum
20	Scott	Stevens	400	01-07-2010	REP	USA	USA	8745.5	Scott Stevens
21	Anna	De Smet	5	01-03-2011	REP	USA	USA	2800	Anna De Smet
22	Deer	Anna	019	04-07-2011	REP	USA	USA	1177	Deer Anna
23	Andrea	ML	339	01-04-2011	REP	USA	USA	9147.5	Andrea ML
24	Christa	Stetteland	21	04-04-2011	REP	USA	USA	109914	Christa Stetteland
25	Anna	De Smet	5	01-06-2011	REP	USA	USA	2750	Anna De Smet
26	Mary S.	MacDonald	477	02-06-2011	REP	USA	USA	11742.3	Mary S. MacDonald
27	Paul	Williams	893	09-09-2011	REP	USA	USA	7629	Paul Williams
28	Oliver H.	Bender	252	09-10-2011	REP	USA	USA	11310	Oliver H. Bender
29	James	Scott	52	02-02-2012	REP	USA	USA	11225.8	James Scott
30	Scott	Brown	502	05-05-2012	REP	USA	USA	7300	Scott Brown
31	Tyals	Shels	32	05-02-2012	REP	USA	USA	800000	Tyals Shels
32	Mary	Page	847	11-04-2012	REP	USA	USA	4800	Mary Page
33	Bill	Patner	247	01-06-2012	REP	USA	USA	15000	Bill Patner
34	Paul	Vernon	11	01-01-2012	REP	USA	USA	14000	Paul Vernon
35	Robert	Berati	11	01-01-2012	REP	USA	USA	11125.1	Robert Berati
36	James	Patner	492	04-08-2012	REP	USA	USA	44000	James Patner
37	Scott	Chen	123	21-08-2012	REP	USA	USA	9510.9	Scott Chen
38	Scott	Johnson	285	11-09-2012	REP	USA	USA	8000	Scott Johnson
39	Er	Ernst	118	01-11-2012	REP	USA	USA	9600	Er Ernst
40	Phyllis	Chabane	30	01-01-2013	REP	USA	USA	11000	Phyllis Chabane
41	Chlo	McGregory	820	01-01-2013	REP	USA	USA	7500	Chlo McGregor
42	Mark	Kochensauer	221	02-01-2013	REP	USA	USA	23000	Mark Kochensauer

Bien, notre exemple n'est toutefois pas complet, il faut ajouter une possibilité de recherche. Revenons donc dans Delphi, et effectuons un double-clic sur le bouton que nous avons posé précédemment, afin d'implémenter l'action de ce dernier. Il suffit alors de saisir ces quelques lignes de code :

```

1 procedure TIWForm1.btn1Click(Sender :
  TObject);
2 begin
3   qryEmployees.Active := False;
4   qryEmployees.Filter := 'full_name like
  ' + QuotedStr('%' + iwdt1.Text +
  '%');
5   qryEmployees.Filtered := True;
6   qryEmployees.Active := True;
7 end;

```

Voilà, exécutons l'application. Il est à présent possible de faire une recherche par nom sur notre liste d'employés.

À présent notre premier projet est terminé. On peut voir qu'en très peu de temps, et avec seulement quelques lignes de code, nous avons réussi à faire un petit programme web fonctionnel. Certes il n'est pas très élaboré d'un point de vue esthétique, mais nous allons voir dans les pages qui suivent comment faire de belles applications web en utilisant tous les concepts connus et standards du web, tout en continuant de coder avec Delphi.

été créées pour notre projet. Il s'agit de :

- DatamoduleUnit.pas (créée uniquement si on choisit l'option « DataPoolConnection ») ;
- ServerController.pas ;
- UserSessionUnit.pas.

#### 4.1 ServerController.pas

Cette unité contient une classe TIWServerController qui va regrouper tous les paramètres du serveur que nous développons. Nous reviendrons plus tard en détail sur cette unité particulière.

#### 4.2 UserSessionUnit.pas

Nous trouvons ici une classe TIWUserSession. Une instance de cette classe est créée pour chaque utilisateur connecté, on peut donc y mettre tous les objets et variables nécessaires à la session. Un *dfm* est également créé, on peut donc y déposer des composants non visuels comme une connexion à une base de données par exemple.

#### 4.3 Datamoduleunit.pas

Cette unité n'est créée que lorsque l'option « Data pool Connection » a été cochée. C'est un *datamodule* classique, que tous les Delphinautes connaissent. Il sert donc exactement à la même chose.

La différence se situe au fait que le serveur va gérer les instances de ce *datamodule* et les fournira à la demande. L'instance fournie sera prise dans un pool de *n* instances, *n* étant paramétrable via la propriété « PoolCount » du composant TIWDatamodulePool dont une instance a été automatiquement ajoutée au TIWServerController lors de la création du projet.

### 5 Le ServerController

Cette unité contient une classe TIWServerController qui, comme son nom le suggère, va permettre de contrôler le serveur et son environnement.

C'est avec cette classe que nous allons pouvoir :

- fixer le port d'écoute de notre application. Attention : ceci n'est pas vrai si vous choisissez de ne pas télécharger la version *bundle* sur le site d'Atozed puisque dans ce cas le port est attribué de façon aléatoire au démarrage du serveur ;
- déterminer le temps d'inactivité durant lequel la session utilisateur reste en mémoire. Passé ce laps de temps (exprimé en minutes) tout appel à la session provoquera un message d'er-

reur de type *Session Timeout* ;

- définir le mode de gestion du numéro de session à l'aide de la propriété « AllowMultipleSessionPerUser ». Celle-ci est à *false* par défaut. Dans ce cas, le numéro de session est stocké dans un *cookie*. Par conséquent lorsqu'une session est démarrée, toute nouvelle instance de navigateur sur un même poste utilisera la même session. Si elle est cochée, alors le numéro de session est passé dans l'URL, on peut donc avoir autant d'instances d'application que de navigateurs ou d'onglets.

Nous détaillerons les autres principales options de cette classe en annexe.

### 6 Les sessions

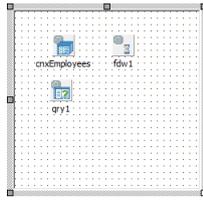
Un des gros atouts d'**Intraweb** est la gestion intégrée des sessions. C'est la classe TIWUserSession de l'unité UserSessionUnit.pas qui va être chargée de cela. Une instance de cette classe est créée pour chaque session. Elle possède peu de propriétés, par contre elle vient avec un *dfm*, comme un DataModule. On peut donc y déposer des composants non visuels dont on a besoin pour la session, comme une connexion à la base de données, un composant Report (Rave, FastReport...).

La gestion à proprement parler se fait à partir de l'objet IWServerController. Dans cet objet, on va notamment pouvoir définir :

- son délai de « TimeOut », c'est-à-dire le délai **d'inactivité** à partir duquel la session est considérée comme obsolète et est détruite ;
- la gestion de son ID. Chaque session possède un ID unique. Ce dernier peut être géré soit par l'URL (dans ce, cas il transite avec chaque appel serveur), soit par *cookie* (il est à ce

moment-là invisible de l'utilisateur). Une autre différence entre les deux modes est que lorsqu'elle est gérée par *cookie*, on ne peut avoir qu'une seule session active par poste client, puisque tous les navigateurs vont lire le même *cookie*.

Maintenant, reprenons notre petite application. Nous n'avons utilisé que l'unité *ufrmMAin* sur laquelle nous avons posé tous les composants. Nous allons réorganiser cela afin de pouvoir réutiliser certains composants dans de nouveaux formulaires. Pour cela, ouvrons à nouveau l'unit *ufrmMain* et nous sélectionnons les composants *TFDConnection*, *TFDWaitcursor*, *TFDquery* puis nous les déplaçons sur le *dfm* de l'unité *IWUserSessionUnit*. Nous avons donc à présent quelque chose qui ressemble à cela :



Revenons sur notre formulaire.

Le correcteur de syntaxe nous indique qu'il existe des erreurs, ce qui est normal puisque nous avons enlevé des composants. Pour supprimer ces erreurs, commençons déjà par ajouter les références nécessaires dans la clause *uses* :

```
1 uses
2   UserSessionUnit, ServerController;
```

Puis, *QryEmployees* faisant maintenant partie de l'objet *UserSession*, il faut modifier la méthode de recherche ainsi :

## 7 Le pool de connexions

Nous avons vu lors de la création d'un projet IntraWeb qu'il est possible de demander un pool de connexions. Cette fonctionnalité va permettre de partager *n* instances d'un *TDatamodule* entre tous les utilisateurs, chaque instance n'étant bien entendu utilisée que par un seul utilisateur à la fois. L'intérêt de ceci est de pouvoir disposer d'un *Datamodule* déjà créé (s'il en existe un libre dans le pool), ce qui va améliorer les performances.

Lorsque cette option est sélectionnée, le *template* de projet rajoute un composant sur le *ServerController*, il s'agit d'un *TIWDataModulePool*. Il va permettre le paramétrage du pool et son activation. Le paramétrage est on ne peut plus simple puisqu'il consiste à indiquer via la propriété *PoolCount* combien d'instances de *TDatamodule* peuvent être contenues dans le pool.

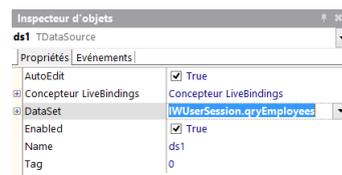
Il dispose également de deux événements :

- *OnCreateDatamodule* : généré lors de l'initialisation du pool, il permet la création de *n* instances du *Datamodule* ;
- *OnFreeDatamodule* : permet de libérer les instances contenues lors de la désactivation du pool ou lors de sa destruction.

Retrouvez la suite de l'article de **Dany Leblanc** en ligne : [lien 34](#)

```
1 procedure TIWForm1.btn1Click(Sender :
2   TObject);
3 begin
4   UserSession.qryEmployees.Active :=
5     False;
6   UserSession.qryEmployees.Filter := '
7     full_name like ' + QuotedStr('%' +
8     idwt1.Text + '%)';
9   UserSession.qryEmployees.Filtered :=
10    True;
11   UserSession.qryEmployees.Active :=
12    True;
13 end;
```

Il ne reste plus qu'à modifier le lien *Dataset* du *TDataSource* et le tour est joué :



Deux fonctions sont également ajoutées au *ServerController*, ce sont deux fonctions globales :

- *LockDatamodule* : cette fonction retourne une instance du *Datamodule*, en la prenant parmi celles disponibles dans le pool. Remarque : si le pool est plein et qu'aucune instance n'est libre, alors une exception est levée ;
- *UnlockDatamodule* : cette méthode permet à une session de libérer le *Datamodule* qu'elle s'est octroyé à l'aide de la méthode *LockDatamodule*. Après l'appel de cette fonction, il ne sera pas détruit, mais sera placé dans le pool en statut disponible pour une autre session qui en aurait besoin.



Bien entendu, ce principe possède des avantages indéniables ; il ne faut cependant pas oublier qu'on récupère une instance qui vient d'être utilisée, il faut donc faire attention à l'état des composants et prendre pour principe, soit de les initialiser au moment où on prend possession de l'instance, soit de les réinitialiser avant de libérer l'instance une fois qu'on en a plus besoin.



# Python

## Les derniers tutoriels et articles

# Concepts Python avancés

## Packaging Python et PyPI

Il vous est sûrement arrivé de vous dire que si tous les modules/packages étaient stockés de manière centralisée, alors cela simplifierait grandement vos déploiements. Eh bien la fondation Python a eu la même réflexion, et a commencé à travailler pour vous.

### 1 Introduction

Pour une partie d'entre nous, s'est déjà posée la question du déploiement : comment s'organiser, comment gérer ses versions... Autant de questions sans forcément de réponses évidentes. La fondation Python a pensé à nous en créant PyPI (prononcer *païpi*) : lien 35.

Le but est fort simple : standardiser les procédures et nous fournir de véritables outils et moyens de s'organiser tant pour le stockage que pour le déploiement.

À travers cet article, je vous invite à découvrir PyPI, son outil dédié Pip, et bien entendu comment créer vos paquets au format officiel, les « .whl ».



Puisque nous allons aborder le packaging dans ce tutoriel, un petit rappel rapide sur les notions de package et de module.

Un package Python n'est concrètement qu'un dossier contenant un fichier « `__init__.py` », fichier qui est vide la plupart du temps. Ce fichier permet juste à Python de savoir qu'il a affaire à un package Python et non à un simple dossier.

Un module, lui, est un simple fichier python, contenant du code simple, une/des classe(s), ou un mélange.

Un package peut contenir des sous-packages et/ou des modules. Pour de plus amples informations, je vous renvoie ici : lien 36.

### 2 Principe

PyPI, pour PYthon Package Index, est en quelque sorte un serveur centralisé de paquets Python. Il centralise les différents modules/paquets mis à disposition par la communauté et en gère les différentes versions.

Si vous ne disposez pas toujours d'une connexion Internet, vous avez également la possibilité de créer une copie en local.

Mieux encore, si pour diverses raisons, vous ne désirez pas utiliser le PyPI officiel, vous pouvez vous créer un PyPI local totalement ou partiellement

vierge et l'utiliser de manière classique.

PyPI ne vient cependant pas seul, puisqu'un outil dédié existe : Pip. Cet outil va permettre d'établir une connexion entre un ordinateur et le serveur PyPI afin d'installer/mettre à jour/désinstaller un module/package avec plus ou moins d'options que nous verrons ensemble.

L'outil Pip va vous permettre d'installer des modules/paquets depuis le serveur PyPI officiel, en ligne, depuis un serveur PyPI local, ou encore depuis un dossier spécifique que vous stipulerez.

### 3 L'outil Pip

Pip est un outil en ligne de commandes, disponible aussi bien pour la branche 2 que pour la branche 3 de Python, sous Linux, Windows et Mac.

Il est normalement inclus par défaut à partir de Python 2.7.9 (pour la branche 2) et Python 3.4 (pour la branche 3).

Pour toutes les autres versions, il vous faudra l'installer via le script get-pip : lien 37. Une fois ce dernier téléchargé, il vous suffira de lancer la commande :

```
1 python get-pip.py
```



Dans certains cas, et malgré que votre version de Python soit éligible à disposer nativement de Pip, ce dernier peut nécessiter des manipulations manuelles d'installation.



Il existe une version de Pip par version de Python. Ainsi, si vous avez sur votre poste un Python 2.7.9 et un Python 3.4, il existera deux Pip sur votre système. La distinction se fait alors de la manière suivante : Pip<version\_en\_deux\_digits> ...

```
1 pip2.7 install 2mp3
2 pip3.4 install 2mp3
```

Dans d'autres cas, la distinction se fera exclusivement de la façon suivante :

```
1 pip install 2mp3
2 pip3 install 2mp3
```

#### 3.1 Le format requirement

Le format requirement est un format très simple permettant de pouvoir stipuler, outre le nom d'un paquet, la version attendue. Les formes possibles d'écriture sont les suivantes :

```
1 NOM
2 NOM==version
3 NOM>=version
4 NOM<=version
5 NOM>version
6 NOM<version
```

Vous avez la possibilité de mettre un ensemble de paquets dans un fichier dit de requirement. Dans ce cas, chaque ligne ne contiendra le nom que d'un seul paquet, lequel devra être saisi au format requirement. Ce fichier porte alors en général le nom de « requirements.txt ».

```
1 ...
2 Pillow>=2.7.0
3 piston-mini-client<=0.7.5
4 psutil==2.1.1
5 pycrypto==2.6.1
6 pycups==1.9.72
7 pycurl==7.19.5
8 pygobject==3.14.0
9 ...
```

#### 3.2 Commandes et options

Comme tout outil en ligne de commande, Pip possède des commandes et des options. Nous allons voir ici les essentielles.

Avant toute chose, rappelons que Pip fonctionne de la façon suivante

```
1 pip <commande> [option] paquet
```

Commandes	
<b>install</b>	Permet d'installer un paquet
<b>uninstall</b>	Permet de désinstaller un paquet
<b>freeze</b>	Permet d'afficher, ou de créer un fichier contenant l'ensemble des paquets installés, au format requirement
<b>list</b>	Permet de lister les paquets installés sur la machine
<b>show</b>	Permet d'afficher des informations à propos d'un paquet donné
<b>search</b>	Permet de rechercher un paquet, selon un mot clé, sur PyPI (local ou non)
<b>wheel</b>	Crée un fichier Wheel à partir du fichier requirement indiqué (nécessite parfois un paquet supplémentaire)
<b>help</b>	Permet d'afficher l'aide

Options	
<b>-h / -help</b>	Permet d'afficher l'aide
<b>-v / -verbose</b>	Plus verbeux
<b>-V / -version</b>	Permet d'obtenir la version de Pip
<b>-q / -quiet</b>	Moins verbeux
<b>-log-file &lt;path&gt;</b>	Redirige la sortie écran vers un fichier de log. Actif par défaut dans le chemin communiqué par l'aide
<b>-log &lt;path&gt;</b>	Inactif par défaut
<b>-proxy &lt;proxy&gt;</b>	Permet de stipuler un proxy à utiliser, sous la forme [utilisateur :mot_de_passe@]proxy.server :port
<b>-timeout &lt;sec&gt;</b>	Définit le timeout pour le socket (par défaut à 15s)
<b>-exists-action &lt;action&gt;</b>	Action par défaut quand un chemin existe déjà : (s)witch (i)gnore (w)ipe (b)ackup
<b>-cert &lt;path&gt;</b>	Permet de définir le chemin vers des certificats alternatifs
<b>-r / -requirement</b>	Permet d'installer/désinstaller l'ensemble des paquets listés dans un fichier requirement
<b>-y / -yes</b>	Permet de ne pas avoir à confirmer une désinstallation



Plus d'options sont disponibles via un « man pip ».



Libre à vous de supprimer des lignes dans le fichier si vous estimez qu'un des paquets n'a rien à faire dans la liste. De même vous pouvez modifier la version attendue.

Ainsi, si nous désirons savoir quels paquets sont installés sur notre machine, nous taperons :

```
1 pip list
```



list et freeze produiront approximativement la même sortie, mais en utilisant un format d'affichage différent.

Si nous désirons chercher les paquets ayant un lien avec les mp3, nous taperons :

```
1 pip search mp3
```



Lors de l'utilisation de l'outil Pip, vous pouvez parfois voir une ligne vous stipulant qu'une nouvelle version est à jour. N'hésitez alors pas à installer cette nouvelle version afin de bénéficier d'une meilleure stabilité et de potentielles nouvelles options.

### 3.3 Lister les paquets installés dans un fichier

Comme vu dans le récapitulatif des commandes et options, vous avez la possibilité de lister l'ensemble des paquets installés. Cependant, il faut savoir que vous pouvez générer un fichier grâce à cela :

```
1 pip freeze > ./requirements.txt
```

Vous disposez alors d'un fichier dit de requirement, souvent nommé requirements.txt. Ce fichier vous permettra ensuite de recréer aisément un environnement similaire.

### 3.4 Installer un groupe de paquets grâce à un fichier requirement

Une fois que vous disposez d'un fichier requirement, le déploiement s'avère très facile :

```
1 pip install -r requirements.txt
```

Pip va alors installer l'ensemble des paquets attendus. Vous serez averti de potentielles erreurs sur votre terminal. Cas échéant, si de nombreux paquets, je vous invite à afficher la liste des paquets installés, à la fin de l'installation, afin de vous assurer que tout a bien été installé.

### 3.5 Installer un paquet depuis un fichier

Pip permet d'installer des paquets disponibles sur Internet, tel que nous venons de le voir, mais aussi des fichiers locaux. Comment ? Tout simplement comme suit :

```
1 pip install ./a_installer/mon_paquet.whl
```

### 3.6 Installer des paquets depuis un dossier

Pouvoir installer une archive est fort pratique, mais si vous en avez 25 à installer ? Bien entendu, vous pourriez saisir 25 fois la même ligne. Mais là aussi, Pip à la solution :

```
1 pip install -f ./mon_dossier
```

Cela installera tous les paquets qui seront trouvés dans « mon\_dossier ».

### 3.7 Installer des paquets depuis Mercurial

Cerise sur le gâteau, Pip sait gérer les gestionnaires de versions (lien 38) Mercurial, Git, Svn, et Bazaar .

Nous allons voir ici Mercurial. Dans le cas de ce dernier, Pip supporte les chemins en http, https, static-http et ssh.

```
1 [-e] hg+http://hg.mon_projet.org/
   MonProjet
2 [-e] hg+https://hg.mon_projet.org/
   MonProjet
3 [-e] hg+ssh://hg.mon_projet.org/
   MonProjet
```

Outre le fait de pouvoir récupérer la dernière révision en date, vous pouvez également en choisir une en particulier, grâce au symbole « @ », en stipulant un hash de révision, un numéro de révision, le nom d'un drapeau, ou bien encore le nom d'une branche précise :

```
1 [-e] hg+http://hg.mon_projet.org/
   MonProjet@da39a3ee5e6b
2 [-e] hg+http://hg.mon_projet.org/
   MonProjet@2019
3 [-e] hg+http://hg.mon_projet.org/
   MonProjet@v1.0
4 [-e] hg+http://hg.mon_projet.org/
   MonProjet@special_feature
```

## 4 Wheel : le nouveau format officiel

Après avoir autorisé différents formats (tarball, egg, zip...) la fondation Python, via la PEP427 (lien 39), définit un format officiel : le wheel.

Une grande vague de migration est actuellement en cours (lien 40), afin de migrer les principaux pa-

ckages vers ce format.

Au cours de cet article, l'ensemble de nos exemples et explications tourneront donc exclusivement autour de ce format, dont les fichiers portent l'extension « .whl ».

## 5 Packaging Python

Voici venir un des moments importants dans la vie de tout développeur Python : construire son propre package.

Nous allons voir ici plus précisément les différents fichiers nécessaires, ainsi que leur contenu, et comment s'en servir pour construire votre paquet.

Voyons maintenant les principaux paramètres que vous pouvez communiquer à la fonction setup : lien 43.



Pour tout paramètre prenant une simple chaîne de caractères, vous êtes limité à 200 caractères.

### 5.1 Prérequis

Afin de pouvoir packager correctement un paquet Python, vous aurez préalablement besoin d'installer les packages setuptools ainsi que wheel.

```
1 pip install setuptools
2 pip install wheel
```

### 5.2 Le fichier setup.py

Il existe différents outils permettant de construire un paquet à partir d'un fichier setup.py. L'outil officiel est distutils, actuellement en cours de remplacement.

L'outil le plus utilisé reste cependant setuptools (lien 41) (doc officielle ici : lien 42). C'est cet outil que nous allons utiliser ici.

Le fichier setup.py est le fichier indispensable pour toute distribution de paquets en Python. En effet, c'est lui qui va donner à Python les instructions indispensables à la construction, tels le nom, la version, le ou les auteur(s)...

Le fichier setup.py peut contenir du code, mais en général l'appel à la fonction « setup » suffit.

#### 5.2.a Spécifier le nom et la version du programme

Les premiers éléments, indispensables, à renseigner sont le nom du package à générer, ainsi que sa version.

Ces deux paramètres sont de simples chaînes de caractères.

```
1 from setuptools import setup
2 setup(name = 'nom de mon package',
3       version = '0.0.1'
4       )
```

Notez que vous pouvez aussi importer la version et/ou le nom depuis votre package, si vous avez défini la variable `__version__`, en l'important dans votre fichier setup.py.

```
1 from setuptools import setup
2 import mon_package
3 setup(name = 'nom de mon package',
4       version = mon_package.__version__
5       )
```

### 5.2.b Informations relatives aux auteurs et mainteneurs

Ici, vous allez pouvoir déclarer qui sont les auteurs et les mainteneurs, ainsi que leurs mails respectifs. Là aussi, il s'agit de simples chaînes de caractères.

```

1 from setuptools import setup
2 import mon_package
3 setup(name = 'nom de mon package',
4       version = mon_package.__version__,
5       author = 'Developpez Dotcom',
6       author_email = 'developpez@dotcom.fr',
7       maintainer = 'Developpez Dotcom',
8       maintainer_email = '
          developpez@dotcom.fr'
9 )

```

### 5.2.c Spécifier des mots clés et catégories pour les recherches

Si jamais vous chargez votre package sur PyPI, par la suite, il sera intéressant de pouvoir effectuer aisément une recherche par mot clé ou par catégorie.

Pour cela, il existe deux paramètres : classifiés et keywords.

Le premier est une liste de strings. Chaque string correspondra à un type prédéfini, dont la liste complète est disponible ici : [lien 44](#).

Keywords est une simple chaîne de caractères dont les mots clés sont séparés par un espace.

```

1 from setuptools import setup
2 import mon_package
3 setup(name = 'nom de mon package',
4       version = mon_package.__version__,
5       keywords = 'exemple documentation
6                 setup.py'
7       classifiés = ['Topic :: Education
8                   ',
9                   'Topic ::
10                  Documentation']

```

### 5.2.d Spécifier les URL relatives au code

Côté URL, vous pouvez spécifier une URL où trouver des informations relatives au code (page du projet, documentation, article de journal, page PyPI...), ainsi qu'une adresse pointant directement vers un lien permettant de charger votre package. Dans les deux cas, il s'agira de simples chaînes de caractères.

```

1 from setuptools import setup
2 import mon_package
3 setup(name = 'nom de mon package',
4       version = mon_package.__version__,
5       url = 'http://mon_projet.org',
6       download_url = 'http://packages.python.org/mon_projet'
7 )

```

### 5.2.e Spécifier les packages et modules à empaqueter

Indispensable, il faut communiquer à la fonction setup les packages et/ou modules à empaqueter.

Le paramètre de base pour cela est « packages ». Il s'agit d'une liste de chaînes de caractères.

```

1 from setuptools import setup
2 import mon_package
3 setup(name = 'nom de mon package',
4       version = mon_package.__version__,
5       packages = ['mon_package']
6 )

```

Cependant, admettons que pour une raison donnée, votre package à empaqueter ne se situe pas directement au même niveau que votre setup.py. Comment faire ? Eh bien là, existe un autre paramètre : « package\_dir » qui est un dictionnaire, où la clé est le nom du package, et la valeur le chemin du dossier où trouver le package (relatif à l'emplacement du setup.py).

```

1 from setuptools import setup
2 import mon_package
3 setup(name = 'nom de mon package',
4       version = mon_package.__version__,
5       package_dir = {'mon_package' : '
6                       les_packages_maison'}
7 )

```

Ici, Python s'attendra à trouver « ./les\_packages\_maison/mon\_ ». Il tombera en erreur sinon.

Cette technique permet de stipuler au cas par cas où trouver les packages. Mais admettons maintenant qu'ils se trouvent tous au même endroit. Dans ce cas, il existe encore une autre façon de faire.

```

1 from setuptools import setup
2 import mon_package
3 setup(name = 'nom de mon package',
4       version = mon_package.__version__,
5       packages = ['mon_package'],
6       package_dir = {'' : '
7                       les_packages_maison'}

```

Dans ce dernier exemple, Python ira toujours chercher les packages stipulés avec le paramètre « packages » dans « ./les\_packages\_maison ».

Il y a également plus simple que tout cela, sous réserve que vos packages soient placés à un niveau inférieur de votre setup.py.

En effet, le package setuptools ne vient pas juste avec setup. Vous y trouverez également des fonctions telles que « find\_packages() ».

Cette fonction vous permettra de détecter automatiquement, de façon récursive, tout package depuis votre position courante, avec la possibilité d'exclure certains packages.

```

1 from setuptools import setup,
2   find_packages
3 import mon_package
4 setup(name = 'nom de mon package',
5       version = mon_package.__version__,
6       packages = find_packages(exclude=[
7           '*tests.*'])

```

```

6     packages_dir = {'' : '
7         les_packages_maison'}

```

Ici, nous demandons à ne pas inclure les niveaux contenant « tests » dans le chemin de modules (ex. : `mon_package.tests.demo` ne sera pas inclus, mais `package.dvp` oui).

Maintenant, imaginons que vous n'avez que des modules et non des packages (ou bien des modules en sus des packages). Le paramètre pour cela est `py_modules`. Il s'agit d'une liste de chaînes de caractères où l'on précise le nom des modules sans l'extension.

```

1 from setuptools import setup
2 import mon_package
3 setup(name = 'nom de mon package',
4       version = mon_package.__version__,
5       packages = ['mon_package'],
6       py_modules = ['test_demo']
7     )

```

### 5.2.f Décrire l'objectif du code

Afin de comprendre ce que fait votre code, il est plus aisé d'ajouter des descriptions. `Setup.py` prend en charge deux types de description : une courte et une longue.

La description courte (en général une phrase) est celle qui est affichée dans PyPI lors d'une recherche. Elle doit décrire brièvement le but du package.

La description longue, elle, peut contenir des informations sur la licence, des exemples, des liens Internet... Pour cette dernière, il est recommandé de disposer d'un fichier texte `README`, placé au même niveau que `setup.py`, et qu'on ira lire.

```

1 from setuptools import setup
2 import os
3 import mon_package
4 setup(name = 'nom de mon package',
5       version = mon_package.__version__,
6       description = 'phrase courte
7         expliquant le but de mon
8         projet',
9       long_description = open(os.path.
10        join(os.path.dirname(__file__
11        ), 'README')).read()
12     )

```

### 5.2.g Spécifier la licence

Bien que vous puissiez spécifier, dans les classifieurs, une licence type, à des fins de recherches, il existe également un paramètre permettant d'indiquer par une simple chaîne de caractères la licence utilisée.

```

1 from setuptools import setup
2 import os
3 import mon_package
4 setup(name = 'nom de mon package',
5       version = mon_package.__version__,
6       licence = 'GPL V3'
7     )

```

### 5.2.h Spécifier les plateformes

Il s'agit là d'un paramètre rarement utilisé. Il s'agit d'une simple chaîne de caractères visant à informer des plateformes pouvant utiliser le package.

Les valeurs courantes utilisées sont ALL, WINDOWS, LINUX, MAC.

```

1 from setuptools import setup
2 import os
3 import mon_package
4 setup(name = 'nom de mon package',
5       version = mon_package.__version__,
6       plateformes = 'ALL'
7     )

```

### 5.2.i Spécifier les dépendances

Vous avez la possibilité de stipuler les dépendances de votre package. Dans `setup.py`, elles peuvent être de trois types différents :

- disponibles dans PyPI;
- non disponibles dans PyPI;
- Autre.

Si le package dont vous dépendez se trouve dans PyPI, il faut utiliser « `install_requires` ». Les paquets sont alors à stipuler au format `requirement`, dans une liste de strings :

```

1 from setuptools import setup
2 import os
3 import mon_package
4 setup(name = 'nom de mon package',
5       version = mon_package.__version__,
6       install_requires = ['docutils>=0.3
7         ', 'numpy-stl>=1.0.0,<=1.6.2']
8     )

```

Si les paquets ne sont pas dans PyPI, il faut qu'ils soient disponibles au format `tar.gz/egg/whl`, en tant que simple fichier `.py` (en local, réseau, ou sur le net), ou disponibles sur un dépôt (Mercurial (`hg+URL`), Git (`git+URL`), ou SVN (`svn+URL`)).

Il faut alors utiliser « `dependency_links` ». Il s'agit là aussi d'une liste de strings.

```

1 from setuptools import setup
2 import os
3 import mon_package
4 setup(name = 'nom de mon package',
5       version = mon_package.__version__,
6       dependency_links = ['http://peak.
7         telecommunity.com/snapshots/']
8     )

```

Enfin, par autre, nous entendons, toutes dépendances qui ne seraient pas totalement obligatoires. Entendez par là, une fonctionnalité spécifique à votre paquet dont tout le monde n'aurait pas besoin (ex. : génération de rapport pdf via `reportlab`, ou de documentation ReST). Il faut alors utiliser « `extras_require` ». Cette fois-ci il s'agira d'un dictionnaire. Les clés correspondront à la fonctionnalité, et la valeur sera une liste des paquets attendus.

```

1 from setuptools import setup
2 import os
3 import mon_package
4 setup(name = 'nom de mon package',
5       version = mon_package.__version__,
6       extras_require = {'PDF': ['
7                           ReportLab>=1.2'],
8                           'ReST': ['
9                               docutils>=0.
10                                3']}]

```

### 5.2.j Les fichiers additionnels (images, config...)

Vous avez la possibilité d'empaqueter tout fichier qui serait utile à votre package (par exemple des images) et qui ne serait pas du code informatique. Pour cela, il vous faudra définir une liste de tuples de type (dossier, liste de fichiers). Il vous faudra utiliser le paramètre « data\_files ».

```

1 from setuptools import setup
2 import os
3 import mon_package
4 setup(name = 'nom de mon package',
5       version = mon_package.__version__,
6       data_files = [('dossier_images', ['
7                       sous_dossier_image/image1.png'
8                       ]),
9                    ('dossier_config', ['
10                       cfg1.cfg', 'cfg2
11                       .cfg', 'cfg3.cfg
12                       '])])

```

### 5.3 Le fichier setup.cfg

En plus du fichier setup.py, vous pouvez créer un fichier setup.cfg (lien 45). Ce fichier optionnel, qui vise à terme à remplacer le setup.py, contiendra alors des informations complémentaires utiles pour la construction.

Par exemple, stipuler que le code est compatible branche 2 et 3 de Python :

```

1 [bdist_wheel]
2 universal=1

```

Ou bien encore compatible uniquement avec une version donnée :

```

1 [bdist_wheel]
2 python-tag = py32

```



Les python tags se fabriquent ainsi : py<version sans le point>-none-plateforme>. En général, on se contente de « py<version sans le point> ».

### 5.4 Création du wheel

La création d'un wheel est en soi assez simple. Une fois le setup.py, et potentiellement le setup.cfg,

créés, il vous suffira de saisir les lignes suivantes, depuis le dossier où se trouve votre paquet, afin de créer votre wheel :

```

1 python setup.py test
2 python setup.py bdist_wheel

```

La première ligne sert à tester le setup.py et vous affichera toute erreur qui pourrait être présente dans ce fichier.

À la suite de la seconde commande, trois nouveaux dossiers sont alors créés :

- build ;
- dist ;
- <nom\_projet>.egg-info.

C'est dans « dist » que vous trouverez le paquet généré.

Voilà, vous disposez alors de votre package Python au format standardisé Wheel.



Les exemples présentés ici sont testés sous Linux. Des problèmes semblent survenir sous Windows. En cas de doute, n'hésitez donc pas à lancer une VM ou un dual boot, afin de vous assurer de la source de l'anomalie.



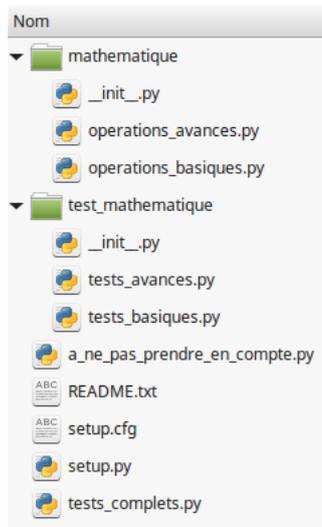
Il semblerait que l'encodage soit parfois mal géré dans certains outils Python. Aussi, si lors du lancement d'une commande, un message lié au caractère ASCII apparaît, commencez par vous assurer que votre chemin ne comporte pas de caractères spéciaux (accents par exemple).

### 5.5 Exemple concret

Afin de clarifier tout ce que nous venons de voir, nous allons prendre un exemple concret.

Pour cela, imaginons que nous avons créé un package basique nommé « mathématique ». Ce package contiendra 2 modules (opérations\_basiques et opérations\_avancées). Un second package, « test\_mathématique » contiendra des modules destinés à tester le package « mathématique ».

Au même niveau que nos packages, nous aurons également un module Python à ne pas prendre en compte, un module Python permettant de lancer l'intégralité des tests sur notre package « mathématique » et bien entendu, nos fichiers standards nécessaires à la création de notre wheel (setup.py, setup.cfg, README.txt).



Voici maintenant le contenu des différents fichiers (les `__init__.py` seront vides, exception faite du `__init__.py` du package « mathematique »).

```

1  __version__ = '0.0.1'

1  def addition(a, b):
2      return a + b
3
4  def soustraction(a, b):
5      return a - b
6
7  def multiplication(a, b):
8      return a * b
9
10 def division(a,b):
11     return a / b
12
13
14 if __name__ == '__main__':
15     print('addition de 1 & 2:', addition
16         (1, 2))
17     print('soustraction de 3 & 2:',
18         soustraction(3, 2))
19     print('multiplication de 2 & 3:',
20         multiplication(2, 3))
21     print('division de 3 par 4:',
22         division(3, 4))

1  def puissance(a, b):
2      return a**b
3
4  def racine_x(a, b):
5      return a**(1./b)
6
7
8  if __name__ == '__main__':
9      print('2 puissance 3:', puissance(2,
10         3))
11     print('racine cubique de 8:',
12         racine_x(8, 3))

1  import sys
2  import os
3  sys.path.append(os.path.join(os.path.
4      dirname(os.path.realpath(__file__)),
5      os.pardir))
6
7  from mathematique.operations_basiques
8  import addition

```

```

6  from mathematique.operations_basiques
7  import soustraction
8  from mathematique.operations_basiques
9  import multiplication
10 from mathematique.operations_basiques
11 import division
12
13 def test_addition():
14     result = addition(1, 2)
15     if result == 3:
16         print('Test addition OK')
17     else:
18         print('Test addition KO')
19
20 def test_soustraction():
21     result = soustraction(3, 2)
22     if result == 1:
23         print('Test soustraction OK')
24     else:
25         print('Test soustraction KO')
26
27 def test_multiplication():
28     result = multiplication(2, 3)
29     if result == 6:
30         print('Test multiplication OK')
31     else:
32         print('Test multiplication KO')
33
34 def test_division():
35     result = division(float(3), 4)
36     if result == 0.75:
37         print('Test division OK')
38     else:
39         print('Test division KO')
40
41 if __name__ == '__main__':
42     test_addition()
43     test_soustraction()
44     test_multiplication()
45     test_division()

1  import sys
2  import os
3  sys.path.append(os.path.join(os.path.
4      dirname(os.path.realpath(__file__)),
5      os.pardir))
6
7  from mathematique.operations_avances
8  import puissance
9  from mathematique.operations_avances
10 import racine_x
11
12 def test_puissance():
13     result = puissance(2, 3)
14     if result == 8:
15         print('Test puissance OK')
16     else:
17         print('Test puissance KO')
18
19 def test_racine_x():
20     result = racine_x(float(8), 3)
21     if result == 2:
22         print('Test racine_x OK')
23     else:
24         print('Test racine_x KO')
25
26 if __name__ == '__main__':
27     test_puissance()
28     test_racine_x()

1  from test_mathematique.tests_basiques
2  import test_addition

```

```

2 from test_mathematique.tests_basiques
  import test_soustraction
3 from test_mathematique.tests_basiques
  import test_multiplication
4 from test_mathematique.tests_basiques
  import test_division
5 from test_mathematique.tests_avances
  import test_puissance
6 from test_mathematique.tests_avances
  import test_racine_x
7
8 if __name__ == '__main__':
9     test_addition()
10    test_soustraction()
11    test_multiplication()
12    test_division()
13    test_puissance()
14    test_racine_x()

```

```

1 Ceci est le contenu du fichier README.
  TXT. Il s'agit d'un fichier texte ou
  RST contenant la description longue
  du paquet. Deporter cette
  description dans un fichier externe
  a setup.py permet d'améliorer la
  lisibilité du code de setup.py.
2 Ce fichier contient normalement au
  minimum le nom du paquet, son but,
  la licence et un hyperlien si
  possible. Il peut être utile d'
  ajouter en sus un exemple.

```

```

1 string1 = 'ceci est un fichier python ne
  contenant que deux strings'
2 string2 = 'a ne pas prendre en compte a
  la creation du package'

```

```

1 from setuptools import setup,
  find_packages
2 import os
3 import mathematique
4
5 setup(name = 'pkg_dvp_mathematique',
6       version = mathematique.__version__
7       ,
8       author = 'dvp',
9       author_email = 'dvp@dvp.com',
10      maintainer = 'dvp',
11      maintainer_email = 'dvp@dvp.com',
12      keywords = 'dvp developpez.com
  deussyss package Python',
13      classifiers = ['Topic :: Education
  ', 'Topic :: Documentation'],
14      packages = ['mathematique', '
  test_mathematique'],
15      description = 'demonstration de
  creation d'un package Python'
16      ,
17      long_description = open(os.path.
  join(os.path.dirname(__file__
  ), 'README.txt')).read(),
18      license = 'GPL V3',
19      platforms = 'ALL',
20      )

```

```

1 [bdist_wheel]
2 universal=1

```

## 6 PyPI officiel

Maintenant que nous avons vu comment packager nos codes, il est temps de voir les interactions

Voilà. Comme vous le constatez, rien de compliqué. Le code se contente juste de mettre à disposition des fonctions, via le package « mathematique », effectuant du calcul relativement simple.

Le package test\_ « mathematique », lui possède deux modules, chacun affecté à un module du package « mathematique », afin d'en tester les fonctionnalités.

Enfin, au plus haut niveau, un module tests\_complets permet de lancer l'intégralité des tests sur le package « mathematique ».

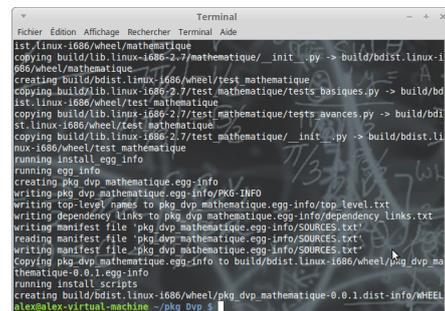
Voilà pour la partie code. À côté, nous retrouvons, un fichier README, deux fichiers setup, et un fichier qu'il ne faudra pas prendre en compte dans notre wheel. Pour ce dernier, il suffit de ne pas y faire référence dans notre setup. Si nous voulions l'embarquer dans le wheel, il suffirait alors de le déclarer en tant que module à prendre en compte.

Cet exemple basique, bien que non exhaustif, va vous permettre de mettre en œuvre les concepts vus précédemment.

Maintenant, placez-vous dans le dossier « Dossier\_de\_packagetage », et lancez la commande :

```
1 python setup.py bdist_wheel
```

Vous disposez alors des trois dossiers cités précédemment. Le dossier nommé dist contiendra votre wheel.



Rendez-vous sur une autre machine ou une VM (notez que vous pouvez également le faire sur le même poste que le packaging), rendez-vous dans le dossier contenant le wheel et exécutez la commande :

```
1 pip install <nom_du_paquet>
```

Et voilà, votre package s'est parfaitement installé. En ouvrant une console Python, vous pouvez désormais faire un « import mathematique ». Puis un help (mathematique) vous communiquera plus d'informations et vous indiquera que le déploiement de votre paquet s'est bien déroulé.

Pour la suite de ce tutoriel, nous nous appuyons sur ce package maison.

ligne.

## 6.1 Ajout d'un paquet

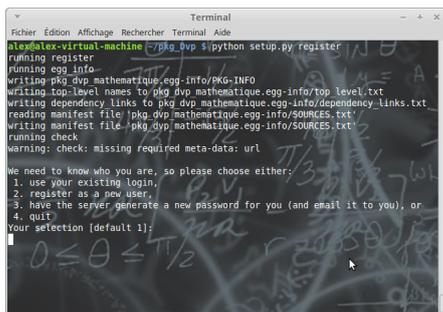
Nous partirons ici des postulats suivants :

- vous disposez d'un paquet fonctionnel ;
- vous disposez du setup.py qui l'accompagne (avec un numéro de version) ;
- le nom du paquet est disponible (doublons non autorisés sur PyPI) ;
- le paquet wheel est installé ;
- le paquet twine est installé (cf. la note plus bas pour l'installation).

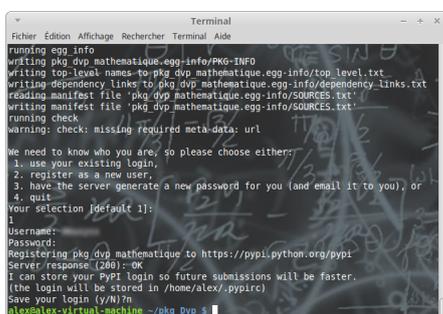
Pour démarrer, ouvrez un terminal, depuis le dossier contenant votre setup.py et tapez :

```
1 python setup.py register
```

Un menu va alors apparaître :



S'il s'agit de votre première connexion à PyPI, il vous faudra d'abord créer un compte. La documentation officielle recommande d'utiliser le formulaire en ligne. Cas échéant, vous pourrez utiliser l'option 2. Si vous possédez déjà un compte, utilisez l'option 1.



Si tout se passe bien, vous aurez alors un message indiquant que l'enregistrement du nom de votre paquet est OK (réponse de type 200 OK).

Enfin, il ne vous reste plus qu'à mettre votre paquet en ligne. Pour cela, nous allons utiliser «twine» : lien 46.

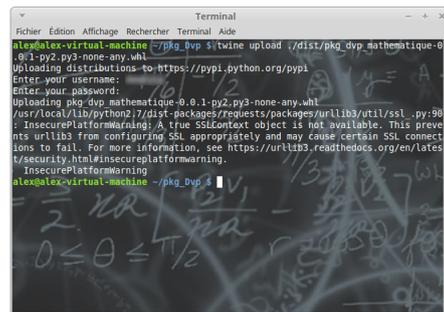
Pour installer twine, il faut saisir la ligne suivante :



```
1 pip install twine pyopenssl ndg-httpsclient pyasn1
```

Ce sont là les paquets minimums. Ensuite, selon votre OS, il se peut que vous deviez ajouter quelques packages supplémentaires.

```
1 twine upload dist/<paquet .whl>
```

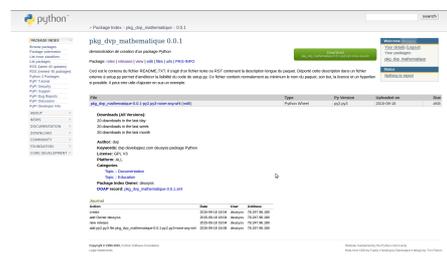


Les plus curieux pourront aller consulter la documentation officielle de la procédure à suivre : lien 47.

Et voilà, notre paquet est en ligne. Si maintenant nous nous connectons à notre compte PyPI, nous avons accès à la page dédiée de notre package :



Comme nous pouvons le voir, beaucoup d'opérations sont alors disponibles en ligne. En cliquant sur view, vous aurez l'aperçu de l'affichage auprès des autres utilisateurs.



Vous pouvez comparer avec [https ://PyPI.python.org/PyPI/pkg\\_dvp\\_mathematique](https://PyPI.python.org/PyPI/pkg_dvp_mathematique).

On peut voir en bas des informations reprises depuis le setup.py.



Pour tout update, veuillez penser à mettre à jour le numéro de version sans quoi l'opération tombera en échec.



Ces derniers mois, l'outil de recherche du site PyPI présentait des défaillances. Aussi, une fois votre paquet ajouté, il est accessible via l'adresse `https://PyPI.python.org/PyPI/<nom_du_paquet>` (lien 48), mais n'apparaît pas forcément lors d'une recherche. De plus l'indexation n'est pas instantanée. Il se peut que vous deviez attendre une semaine avant de voir votre paquet apparaître.

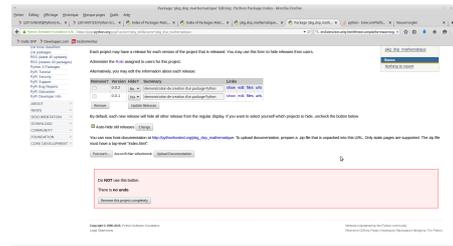
- passer par Pip qui le chargera et l'installera d'office pour vous, si vous le désirez.

### 6.3 Supprimer un paquet

La suppression d'un paquet est manuelle, sur le PyPI. Pour cela, rendez-vous sur le PyPI, puis identifiez-vous.

Ouvrez alors la page du paquet concerné, et choisissez le lien « release ».

Des cases à cocher, et des boutons sont alors à votre disposition pour supprimer tout ou partie du package.



### 6.2 Récupération d'un paquet

Pour récupérer un paquet, vous avez trois solutions principales possibles :

- vous connecter sur la page du paquet, et télécharger le paquet ;
- le télécharger avec un `wget` sous Linux, en ligne de commande ;

## 7 PyPI local

Comme nous venons juste de le voir, avec le PyPI officiel, rien de bien compliqué. Passons maintenant au PyPI local.

Afin de le créer, nous nous baserons sur l'outil `devpi` : lien 49. Il s'agit d'un outil permettant de se créer en local ou sur serveur, un clone du serveur PyPI officiel.

Par la suite, vous disposez également de la possibilité de n'activer votre serveur PyPI qu'en local, ou d'activer une interface réseau afin de la partager sur votre réseau interne, ou sur Internet.



Si vous ne trouvez pas réponse à votre question dans les sections qui suivent, je vous renvoie vers le manuel utilisateur : lien 50. Une annexe est également disponible, en fin de ce tutoriel, afin de résumer les principales commandes DEVPI.

### 7.2 Installation/désinstallation

Pour l'installation rien de plus simple : il est disponible sur PyPI, d'où le besoin d'une connexion Internet.



Il est tout à fait possible d'installer directement `devpi` en ayant téléchargé préalablement son paquet sur PyPI. Pour cela, suivez la procédure décrite dans la section concernant Pip.

Ensuite, dans un terminal, saisissez :

```
1 pip install devpi
2 pip install devpi-server
```

Pour le désinstaller, saisissez :

```
1 pip uninstall devpi
2 pip uninstall devpi-server
```



Assurez-vous de bien disposer des droits administrateur sur votre système.

### 7.1 Prérequis

Pour pouvoir installer PyPI en local, rien de plus simple. Vous devez disposer :

- d'une connexion Internet ;
- de Python ;
- de Pip.

Une fois que vous avez tout cela, vous pouvez envisager l'installation de PyPI en local.

### 7.3 Lancement/contrôle/arrêt du PyPI local

Pour démarrer le serveur PyPI local, rien de plus simple :

```
1 devpi-server --start
```

```

alex@alex-virtual-machine ~$ ./pkg_devpi $ devpi-server --start --host=0.0.0.0
2015-09-18 20:54:31,981 INFO NOCTX Loading node info from /home/alex/.devpi/ser
ver/.nodeinfo
2015-09-18 20:54:31,981 INFO NOCTX generated uuid: 09f6d8f01759441bbdcffcbea14d
3b52
2015-09-18 20:54:31,981 INFO NOCTX wrote nodeinfo to: /home/alex/.devpi/ser
ver/.nodeinfo
2015-09-18 20:54:31,994 INFO NOCTX DB: Creating schema
2015-09-18 20:54:32,010 INFO [Wtx-1] setting password for user u'root'
2015-09-18 20:54:32,010 INFO [Wtx-1] created user u'root' with email None
2015-09-18 20:54:32,010 INFO [Wtx-1] created root user
2015-09-18 20:54:32,010 INFO [Wtx-1] created root/pypi index
2015-09-18 20:54:32,013 INFO [Wtx-1] fswriter0: committed: keys: u',config',u'r
oot',config'
starting background devpi-server at http://0.0.0.0:3141
/home/alex/.devpi/server/.xproc/devpi-server /usr/local/bin/devpi-server --host
=0.0.0.0
process u'devpi-server' started pid=11837
devpi-server process startup detected
logfile is at /home/alex/.devpi/server/.xproc/devpi-server/xprocess.log
alex@alex-virtual-machine ~$ ./pkg_devpi $

```

Le serveur est alors disponible sur `http://localhost:3141` par défaut.

 3141 est une référence au nombre pi (3.141) afin de faire un jeu de mots avec PyPI et devpi.

Par la suite, vous pouvez consulter son état :

```
1 devpi-server --status
```

```

alex@alex-virtual-machine ~$ ./pkg_devpi $ devpi-server --status
2015-09-18 20:56:00,225 INFO NOCTX Loading node info from /home/alex/.devpi/ser
ver/.nodeinfo
2015-09-18 20:56:00,225 INFO NOCTX wrote nodeinfo to: /home/alex/.devpi/ser
ver/.nodeinfo
server is running with pid 11837
alex@alex-virtual-machine ~$ ./pkg_devpi $

```

Ou bien encore, consulter ses logs :

```
1 devpi-server --log
```

```

alex@alex-virtual-machine ~$ ./pkg_devpi $ devpi-server --log
2015-09-18 20:56:35,842 INFO NOCTX Loading node info from /home/alex/.devpi/ser
ver/.nodeinfo
2015-09-18 20:56:35,843 INFO NOCTX wrote nodeinfo to: /home/alex/.devpi/ser
ver/.nodeinfo
Last lines of devpi-server log
2015-09-18 20:54:32,269 INFO NOCTX wrote nodeinfo to: /home/alex/.devpi/ser
ver/.nodeinfo
2015-09-18 20:54:32,272 INFO NOCTX retrieving initial name/serial list
2015-09-18 20:54:35,261 INFO [Wtxd] fswriter1: committed: keys: u'root/pypi/ini
tialLinks'
2015-09-18 20:54:55,405 INFO NOCTX Found plugin devpi-web-2.2.3 (/usr/local/lib
/python2.7/dist-packages)
2015-09-18 20:54:55,469 INFO [NOTI] [Rtx1] Search-Indexing root/pypi:
2015-09-18 20:54:55,467 INFO [MIR] changelog/update tasks starting
2015-09-18 20:54:55,483 INFO NOCTX devpi-server version: 2.1.5
2015-09-18 20:54:55,483 INFO NOCTX serverid: /home/alex/.devpi/server
2015-09-18 20:54:55,483 INFO NOCTX uuid: 09f6d8f01759441bbdcffcbea14d3b52
2015-09-18 20:54:55,483 INFO NOCTX serving at url: http://0.0.0.0:3141
2015-09-18 20:54:55,483 INFO NOCTX bug tracker: https://bitbucket.org/hpk42/dev
pi/issues
2015-09-18 20:54:55,483 INFO NOCTX IRC: #devpi on irc.freenode.net
2015-09-18 20:54:55,483 INFO NOCTX HIT Ctrl-C to quit.
2015-09-18 20:54:55,620 INFO [read] GET
2015-09-18 20:55:22,433 INFO [NOTI] [Rtx1] Committing 66537 new documents to se
arch index.
2015-09-18 20:55:57,098 INFO [NOTI] [Rtx1] Finished committing 66537 documents
to search index.
2015-09-18 20:55:57,094 INFO [NOTI] [Rtx1] finished initial indexing op
logfile is at /home/alex/.devpi/server/.xproc/devpi-server/xprocess.log
alex@alex-virtual-machine ~$ ./pkg_devpi $

```

Enfin, pour l'arrêter, utilisez la commande qui suit :

```
1 devpi-server --stop
```

```

alex@alex-virtual-machine ~$ ./pkg_devpi $ devpi-server --stop
2015-09-18 20:57:19,257 INFO NOCTX Loading node info from /home/alex/.devpi/ser
ver/.nodeinfo
2015-09-18 20:57:19,258 INFO NOCTX wrote nodeinfo to: /home/alex/.devpi/ser
ver/.nodeinfo
killed server pid=11837
alex@alex-virtual-machine ~$ ./pkg_devpi $

```

Bien entendu, il ne s'agit pas là des seules options disponibles pour devpi-server. La documentation officielle ([lien 51](#)) vous informera bien plus. Cependant, voici celles qui pourraient réellement vous être utiles :

Option	Description
<code>-host HOST</code>	Permet de stipuler une adresse autre que localhost pour accéder au PyPI. Pour autoriser un accès autre qu'en local, utilisez « <code>-host=0.0.0.0</code> »
<code>-port PORT</code>	Permet de stipuler un port autre que 3141
<code>-theme THEME</code>	Permet de stipuler un dossier pour personnaliser l'apparence de votre PyPI local (consultez la documentation officielle pour plus d'information : <a href="#">lien 52</a> )
<code>-version</code>	Permet d'afficher la version en cours d'exécution
<code>-export PATH</code>	Permet d'exporter la BDD de devpi-server dans le chemin indiqué
<code>-import PATH</code>	Permet d'importer une BDD devpi-server depuis le chemin indiqué

## 7.4 Concepts basiques

### 7.4.a Serveur

Le serveur devpi fonctionne à la manière de PyPI, mais n'est pas un clone réel de PyPI.

Ce qu'il faut comprendre, c'est que si le fonctionnement avec Pip est similaire (à l'exception que vous devez stipuler expressément l'adresse du serveur à utiliser), en accès web, l'interface graphique n'a rien à voir.

De plus, selon le type d'index, le fonctionnement

diffère également.

Ainsi, si vous utilisez l'index par défaut `/root/.PyPI`, à la demande d'un paquet, devpi va d'abord regarder si le paquet est disponible en mémoire (sur votre installation personnelle). Si ce n'est pas le cas, alors il cherchera à se connecter au PyPI officiel afin de le mettre en cache sur votre machine. Il n'aura ainsi, par la suite, plus à se connecter sur PyPI pour en disposer.

En revanche, dans le cas d'index personnel, s'il vous faudra charger l'ensemble de vos paquets, ces derniers seront directement hébergés en local.

### 7.4.b Index

Les index sont des conteneurs vous permettant de stocker paquets, résultats de tests et documentation.

Un index peut être totalement autonome, ou bien dériver lui-même d'un autre index, que cet index de référence appartienne à l'utilisateur courant ou non. Vous pouvez disposer de plusieurs index.

Cependant, cela va plus loin.

Ces conteneurs possèdent deux principaux modes de fonctionnement : volatil et non volatil. Par volatil, il faut comprendre avoir le droit d'écraser les données par d'autres d'une même version. Un index non volatil ne pourra être effacé ou modifié. La principale utilisation des index non volatils se trouve en production.

Par défaut, à la création, l'utilisateur root dispose d'un index PyPI. Il s'agit d'un index spécial unique. En effet, ce dernier est en lecture seule et aucun upload n'est possible vers cet index.

### 7.4.c Utilisateurs

Les utilisateurs constituent une composante complémentaire aux index.

Prenons l'exemple d'une entreprise. En effet, tous les employés n'ont pas besoin d'avoir accès aux mêmes paquets.

Ainsi, l'intégration nécessitera des paquets spécifiques aux installations sur machines, l'équipe développement, des paquets spécifiques à des mesures physiques...

Trois fonctionnements sont alors possibles :

- tout mettre au même endroit (un utilisateur, un index) (et risquer des conflits de noms en plus du mélange des paquets des différentes équipes) ;
- créer un utilisateur par équipe avec chacun un ou plusieurs index ;
- créer un utilisateur par projet, avec un index ou plusieurs par équipe.

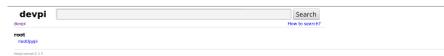
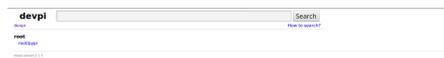
Force est de constater que les deux dernières solutions sont les plus propres. Le compartimentage affiné est rendu possible par l'utilisation conjointe utilisateur/index. En effet, ne disposer que de la possibilité de créer des utilisateurs ou que des index reviendrait à retomber sur la solution 1, avec les désagréments que cela pourrait engendrer.

De plus, afin de sécuriser un peu plus l'accès aux paquets, les utilisateurs disposent (optionnellement) de mots de passe.

## 7.5 Paramétrage principal

Pour effectuer du paramétrage, il vous faudra avant tout définir auprès de devpi, la localisation du serveur, puis indiquer l'index à utiliser :

```
1 devpi use http://localhost:3141
2 devpi use /root/PyPI
```



### 7.5.a Se connecter et se déconnecter

Au cours des diverses manipulations, il vous faudra vous authentifier en tant que root, ou en tant qu'utilisateur. Attention toutefois, car à la création, root ne possède pas de mot de passe :

```
1 devpi login root --password ''
2 devpi logoff
```



Pour la suite des opérations, assurez-vous bien d'être systématiquement identifié. Au minimum en tant que root.

### 7.5.b Ajout/modification/suppression d'un utilisateur

Un utilisateur sera identifié par son pseudonyme, son e-mail, et son mot de passe. L'option « -c » est utilisée pour la création d'un nouvel utilisateur :

```
1 devpi user -c dvp email=dvp@dvp.test
   password=dvp
```

Une fois l'utilisateur créé, vous pouvez contrôler ses paramètres aisément (hors mot de passe) :

```
1 devpi getjson /dvp
```

```

alex@alex-virtual-machine~/pkg_Dvp $ devpi login root --password ''
logged in 'root', credentials valid for 10.00 hours
alex@alex-virtual-machine~/pkg_Dvp $ devpi user -c dvp email=dvp@dvp.test passw
grd=dvp
user created: dvp
alex@alex-virtual-machine~/pkg_Dvp $ devpi getjson /dvp
{
  "result": {
    "email": "dvp@dvp.test",
    "indexes": {},
    "username": "dvp"
  },
  "type": "userconfig"
}
alex@alex-virtual-machine~/pkg_Dvp $

```

```

alex@alex-virtual-machine~/pkg_Dvp $ devpi login dvp --password=dvp
logged in 'dvp', credentials valid for 10.00 hours
alex@alex-virtual-machine~/pkg_Dvp $ devpi index -c dev bases=''
http://localhost:3141/dvp/dev:
type=stage
bases=
volatile=True
uploadtrigger_jenkins=None
acl upload=dvp
pypi whitelist=
alex@alex-virtual-machine~/pkg_Dvp $ devpi index -c prod bases=''
http://localhost:3141/dvp/prod:
type=stage
bases=
volatile=True
uploadtrigger_jenkins=None
acl upload=dvp
pypi whitelist=
alex@alex-virtual-machine~/pkg_Dvp $

```

En cas d'erreur, ou de besoin, vous pouvez modifier l'utilisateur via l'option « -m ». Il vous faudra toutefois vous authentifier auparavant, si cela n'est pas déjà fait :

```

1 devpi login dvp --password dvp
2 devpi user -m dvp email=dvp@developpez.com

```

 Seuls l'e-mail et le mot de passe sont modifiables.

Si l'utilisateur n'a plus de raison d'exister, il suffit de se connecter en root (seul utilisateur ayant les droits adéquats), puis de le supprimer (l'option « -y » est là pour ne pas avoir à confirmer la suppression) :

```

1 devpi user dvp -y --delete

```

Enfin, pour connaître la liste des utilisateurs créés, saisissez la ligne suivante :

```

1 devpi user -l

```

### 7.5.c Définir les droits admin pour la gestion des utilisateurs

Par défaut, tout utilisateur possède le droit de créer d'autres d'utilisateurs. Il est cependant possible d'empêcher cela via l'option `--restrict-modify` :

```

1 devpi -server --restrict-modify root

```

## 7.6 Index

Par défaut, devpi crée un index de PyPI (`/root/-PyPI`), et charge en cache, depuis PyPI tout paquet que vous demanderez dans cet index. Cela nécessite donc une connexion Internet.

### 7.6.a Créer un index vierge

Connectez-vous avec votre user, utilisez la commande adéquate, en spécifiant une base vide :

```

1 devpi login dvp --password dvp
2 devpi index -c dev bases=''

```

### 7.6.b Créer un index héritant d'un autre index

Si toutefois vous désirez non pas un index vierge, mais héritant d'un autre index (bases communes, mais évolutions différentes par exemple), il vous suffit de stipuler le nom de l'index en paramètre, dans « bases » :

```

1 devpi index -c dev_ihm bases=/dvp/dev

```

### 7.6.c Associer un index à un utilisateur

Il est tout à fait possible, depuis l'utilisateur root, de définir des index pour les autres utilisateurs.

Pour cela, il suffit de stipuler le chemin de l'index :

```

1 devpi index -c /dvp/dev bases=''

```

### 7.6.d Supprimer un index

Si vous vous êtes trompé à la création d'un index, si un projet est avorté, ou si pour toute autre raison, vous désirez supprimer un index, la commande est la suivante :

```

1 devpi index -y --delete /dvp/bad_index

```

 L'effacement est non réversible.

```

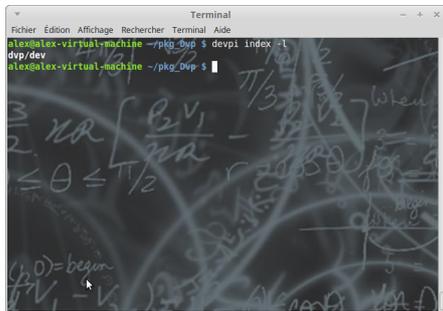
alex@alex-virtual-machine~/pkg_Dvp $ devpi login dvp --password=dvp
logged in 'dvp', credentials valid for 10.00 hours
alex@alex-virtual-machine~/pkg_Dvp $ devpi index -c dev bases=''
http://localhost:3141/dvp/dev:
type=stage
bases=
volatile=True
uploadtrigger_jenkins=None
acl upload=dvp
pypi whitelist=
alex@alex-virtual-machine~/pkg_Dvp $ devpi index -c prod bases=''
http://localhost:3141/dvp/prod:
type=stage
bases=
volatile=True
uploadtrigger_jenkins=None
acl upload=dvp
pypi whitelist=
alex@alex-virtual-machine~/pkg_Dvp $ devpi index -y --delete /dvp/prod
index deleted: <URL 'http://localhost:3141/dvp/prod'>
alex@alex-virtual-machine~/pkg_Dvp $

```

### 7.6.e Afficher la liste des index disponibles

Pour afficher la liste des index disponibles, il suffit d'utiliser la commande :

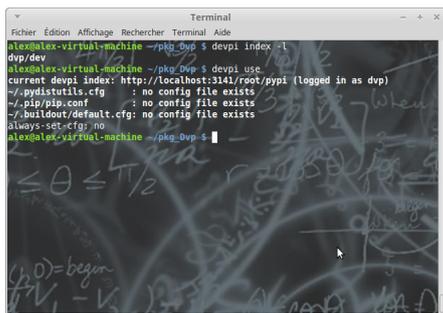
```
1 devpi index -l
```



### 7.6.f Connaître l'index courant

Afin de connaître l'index courant, il suffit de saisir la ligne suivante :

```
1 devpi use
```

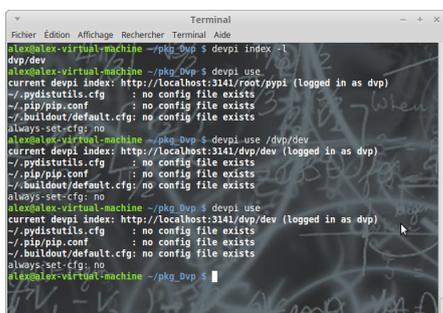


### 7.6.g Définir l'index courant

Si vous disposez de plusieurs index, il peut être utile de définir l'index désiré, afin qu'il devienne l'index courant. Pour cela, utilisez la commande :

```
1 devpi use <indexname>
```

avec <indexname> du type /dvp/nom\_index

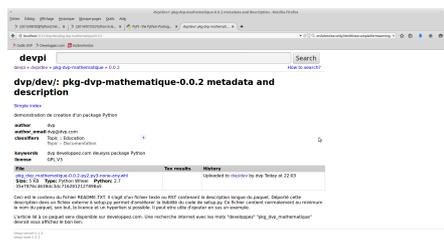
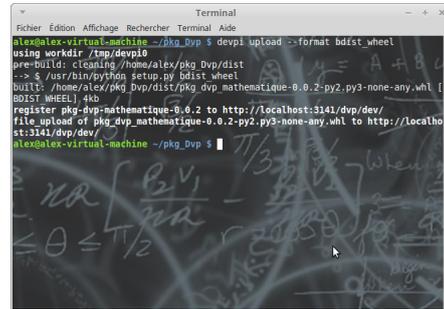


## 7.7 Ajout d'un paquet

Pour uploader un paquet, placez-vous dans le dossier contenant le setup.py. Assurez-vous d'être dans le bon index, puis saisissez :

```
1 devpi upload --format bdist_wheel
```

Le paquet va alors être créé et chargé dans l'index. S'il s'agit d'un nouveau projet, une entrée sera créée dans l'index.

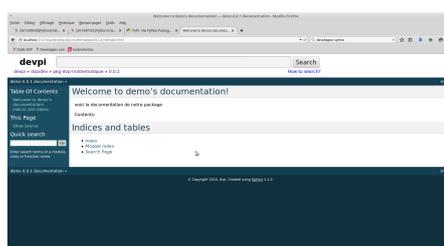
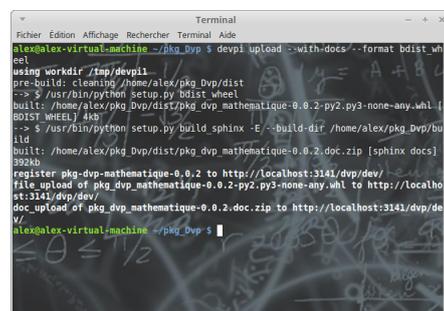


## 7.8 Ajout de documentation Sphinx

Lors d'un upload, vous avez la possibilité de charger une documentation associée à votre package. Pour cela, cette documentation doit être écrite au format Sphinx : [lien 53](#). Cette documentation doit être placée dans un dossier « doc » situé au niveau du setup.py. Il suffira de préciser qu'il faut construire la doc et l'uploader, lors de l'ajout du package :

```
1 devpi upload --with-docs --format bdist_wheel
```

Bien entendu, vous pourrez joindre dans cette documentation tout fichier complémentaire que vous voudriez joindre.



## 7.9 Mise à jour

Pour effectuer une mise à jour, il suffit de réexécuter la même commande que pour un ajout.

## 7.10 Récupération d'un paquet

Dans notre cas précis, nous allons utiliser l'option « -i » de PyPI, nous permettant de lui indiquer l'adresse d'un PyPI différent du PyPI officiel.

```
1 pip install -i http://localhost:3141/
  root/PyPI/ <nom_du_paquet>
```

Bien entendu, si vous désirez une version précise du paquet, cela se passe via un format requirement.

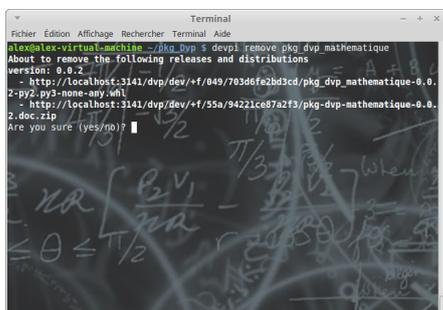
Pour installer un ensemble de paquets listés dans un fichier de requirement :

```
1 pip install -i http://localhost:3141/
  root/PyPI/ -r requirement.txt
```

## 7.11 suppression d'un paquet

La suppression d'un package et des fichiers associés passe par la commande remove. Il ne faudra pas préalablement oublier de définir l'index courant adéquat.

```
1 devpi remove <nom_du_package>
```



Notez que vous pouvez fort bien ne retirer qu'une version donnée via le format requirement :

```
1 devpi remove <nom_du_package>==2.0
```



Si vous supprimez l'intégralité des versions d'un package donné, son entrée dans l'index est automatiquement supprimée.

## 7.12 Navigation dans devpi

Maintenant que nous avons vu comment alimenter notre serveur personnel de packages Python, nous allons voir comment naviguer à l'intérieur de celui-ci.

### 7.12.a Page d'accueil

Rendez-vous sur <http://localhost:3141>



### 7.12.b Afficher un index sous forme de tableau

Puis cliquez sur le lien « /dvp/dev » afin d'ouvrir la page d'accueil de l'index :



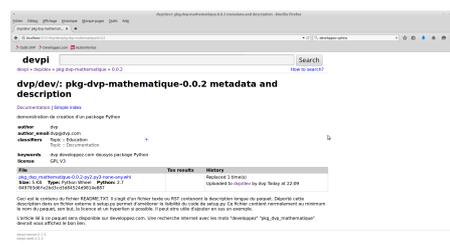
### 7.12.c Afficher un index sous forme de listing

Depuis cette page, vous pouvez changer de type d'affichage, en cliquant sur le lien « simple index ».



### 7.12.d Page d'accueil d'un package donné

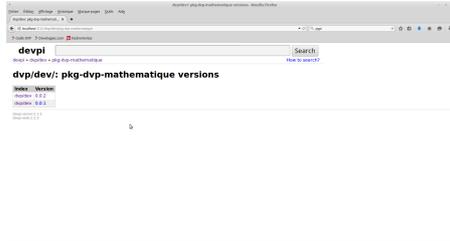
Cliquez sur le nom du package dans le listing, ou dans la colonne « info page », depuis l'index au format tableau.



Cette page correspond à la dernière version.

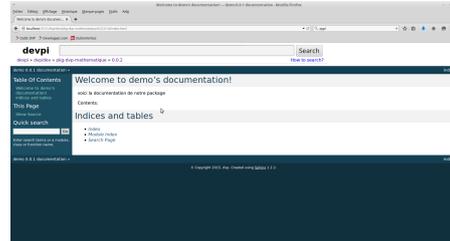
### 7.12.e Afficher l'historique d'un package

Depuis la page précédente, cliquez sur le nom du package, juste sous le champ de recherche, afin d'afficher l'ensemble des révisions disponibles.



### 7.12.f Affichage de la documentation

Depuis la page d'accueil d'une version donnée d'un package, vous pouvez cliquer sur le lien « documentation » afin d'afficher la documentation liée à cette version du package.



Retrouvez la suite de l'article d'*Alexandre Galode* en ligne : [lien 54](#)



# LabVIEW

## Les derniers tutoriels et articles

# « LabVIEW : programmation et applications » ou comment apprendre à utiliser LabVIEW

Cet article est un ensemble d'extraits du livre « LabVIEW, programmation et applications » : lien 55 .  
 Retrouvez le livre sur Amazon : lien 56 !

## 1 Avant-propos : extrait du livre

L'ouvrage est divisé en six chapitres. Le premier présente les concepts et l'intérêt de la programmation graphique. Le flux de données qui est une notion fondamentale à comprendre sous LabVIEW est défini, ainsi que les éléments nécessaires à sa propagation et son exploitation.

Le chapitre 2 permet de décrire les éléments de base de l'environnement de programmation, illustrés avec des exemples simples. Le lecteur apprend à utiliser les objets pour réaliser une interface utilisateur, mais aussi les structures de programmation et les bibliothèques de fonctions. Le langage de développement graphique LabVIEW est intuitif, mais cela ne doit pas faire oublier que, comme avec tous les langages, il faut respecter des règles.

Le chapitre 3 aborde des aspects plus avancés de l'utilisation de l'environnement en définissant des techniques permettant au code d'être maintenable, évolutif, performant et compréhensible par d'autres utilisateurs. Il permet de décrire par des exemples concrets et détaillés les structures de programme à utiliser afin de permettre au lecteur de réaliser une application. L'utilisateur sera capable à la suite de

ce chapitre de préparer la certification LabVIEW développeur.

Les trois derniers chapitres abordent les capacités spécifiques de LabVIEW pour l'acquisition, l'analyse et la présentation des données. Il est le langage de développement le plus efficace pour réaliser un tel système. Dans le chapitre 4, le lecteur apprendra à construire un système de mesure ou de génération de signal, à base de carte d'acquisition (DAQ), d'instrument (IVI, VISA), de système Temps Réel (RT) ou de FPGA. Différentes méthodes de connexion seront envisagées (Série, GPIB, Ethernet, PCI, etc.), ainsi que la réalisation de drivers dans les règles de l'art. Après cette collecte des données, dans le chapitre 5 l'utilisateur pourra à partir de bibliothèques intégrées réaliser un traitement du signal, une analyse mathématique ou un filtrage de l'information. Dans le dernier chapitre, nous illustrerons comment archiver et échanger les données ainsi traitées. Nous verrons comment finaliser son développement, en générant un exécutable autonome et en réalisant un rapport professionnel sous Microsoft Office, pour présenter ses données.

## 2 Introduction

### 2.1 LabVIEW ? Vous avez dit LabVIEW ? Le flux de données ?

LabVIEW est un environnement de développement complet, graphique, compilé et particulièrement bien adapté au domaine de l'acquisition et de la mesure. Son approche totalement graphique offre une souplesse et une dimension intuitive inégalée. Comparativement aux langages textuels, il offre la même puissance de programmation, mais sans le côté abstrait et complexe lié à la syntaxe.

Orienté tests et mesures, il dispose de nombreuses fonctions permettant de piloter facilement des cartes d'acquisition et autres instruments, mais aussi de filtrer, d'analyser et de présenter les données. Ce langage est également appelé code G. Le code est représenté par un schéma composé de fonctions, de structures et de fils qui propagent les données. L'approche visuelle, l'interface entre le code et le développeur s'opposent sur la forme et la philosophie aux langages textuels, comme le C. LabVIEW

est redoutable de puissance et n'a rien à envier aux autres langages.



Une des différences fondamentales de LabVIEW (par rapport aux langages classiques) est que ce langage suit un modèle de **flux de données**, et non de flux d'instructions. Cela signifie que pour un langage textuel, ce sont les instructions qui ont la priorité, alors qu'avec LabVIEW, ce sont les données. **Une fonction s'exécute donc à partir du moment où elle dispose à ses entrées de toutes les données dont elle a besoin.**

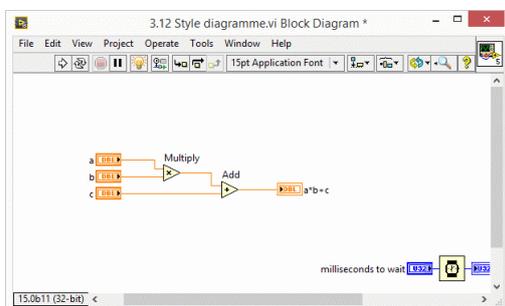
L'exemple suivant, grâce à l'animation, illustre le principe du flux de données. Il réalise un calcul mathématique très simple  $a \times b + c$ .

- La fonction « Multiply » s'exécute lorsqu'elle dispose des données à ses entrées ( $a=1$  et  $b=2$ ), donc dès le départ.
- La fonction « Add », dispose de l'entrée «  $c = 3$  », mais elle est en attente du résultat de la fonction « Multiply ». Dès que la valeur « 2 » est calculée en sortie de la fonction « Multiply », elle devient disponible pour la fonction « Add » qui va pouvoir être calculée.
- La fonction, en bas à droite de l'image, permet d'attendre une seconde. Il est très important de constater que cette attente s'exécute en même temps que la fonction qui calcule  $a \times b + c$ .



**Lorsqu'un langage classique est ainsi séquentiel, LabVIEW est naturellement prédisposé au parallélisme.** Ce qui augmente encore sa puissance et la rapidité d'exécution du code.

La fenêtre qui contient le code est appelée le diagramme. Cette notion sera développée par la suite.



## 2.2 LabVIEW dans le monde de l'instrumentation

LabVIEW est un des premiers langages de programmation graphique destinés au développement d'applications d'instrumentation. Un logiciel d'instrumentation pourrait être défini comme un programme permettant de contrôler depuis un ordinateur, un système allant du capteur à la chaîne d'acquisition ou de commande jusqu'à l'édition du rapport final. Couplé à des cartes d'entrées/sorties, il permet de gérer des flux d'informations numériques ou analogiques et de créer ou de simuler des instruments de mesure (oscilloscope, compteur, multimètre, etc.). Le temps nécessaire à l'assemblage des composants de ce type de système est négligeable par rapport à celui nécessaire à sa programmation dans un langage classique (C, Pascal, Ada, etc.). Les interfaces utilisateur, développées avec ces langages, sont le plus souvent obscures et incompréhensibles. Les utilisateurs disposent avec LabVIEW d'un puissant outil intégré d'acquisition, d'analyse et de présentation des données, une solution qui entraîne un gain notable de productivité comparable à celui obtenu par l'introduction des tableurs dans les logiciels financiers. Pour cela, le langage utilise toute la convivialité des interfaces interactives des ordinateurs actuels en fournissant des objets proches graphiquement des objets réels (voyants, curseurs, interrupteurs, boutons, graphes, etc.), mais aussi des commandes système (pour une représentation plus standard) ainsi que des outils familiers et simples d'utilisation pour la programmation (structures de programmation, fonctions arithmétiques, fonctions logiques, comparateurs, etc.).



## 3 Les bases de la Programmation

Une grande partie du livre (200 pages sur 450) couvre la découverte de l'environnement de développement

intégré LabVIEW ou EDI, puis les bases de la programmation pour pouvoir réaliser des pro-

grammes « simples ».

Il existe de nombreux tutoriels sur Internet sur les bases de la programmation LabVIEW (sur developpez.com ou ni.com). Je fais le choix de restreindre cette « initiation » dans ce tutoriel. Le but est de plus développer les « techniques avancées » qui sont rarement évoquées sur Internet.



Le principe important à comprendre dans l'environnement de développement LabVIEW, est que le langage G ainsi écrit suit un modèle **flux de données**. Une fonction s'exécute à partir du moment où elle dispose en ses entrées de toutes les données dont elle a besoin. LabVIEW est donc prédisposé au parallélisme. Deux fonctions indépendantes s'exécutent en parallèle.

### 3.1 Instrument virtuel (VI) sous LabVIEW

Historiquement, un logiciel de mesure est un instrument de mesure qui est contrôlé depuis un ordinateur à la place des boutons sur sa face-avant.

Cette logique conduit à la notion d'instrument virtuel : instrument réel contrôlé depuis un ordinateur.

Un instrument virtuel est donc un programme qui présente une interface sous forme graphique (IHM) pour l'apparenter à un instrument physique. Les utilisateurs manipulent alors des instruments depuis l'ordinateur (virtuels) comme s'il s'agissait d'instruments physiques sur « étagère » (réels).

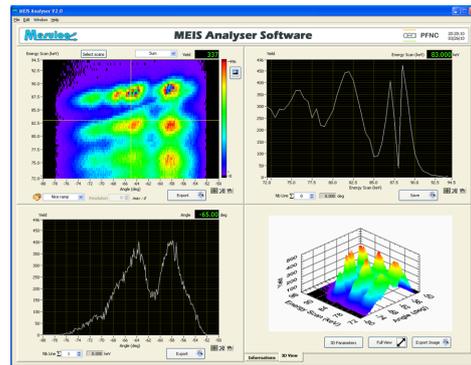


Comme historiquement l'environnement de développement (IDE) est orienté « instrumentation », les inventeurs de LabVIEW ont donné l'extension .vi (Virtual Instrument : VI) au programme développé avec cet environnement pour faire le lien avec le pilotage d'instrument depuis l'ordinateur. C'est donc l'extension du fichier sur le disque « MonCode.vi », comme un document Word est de type « MonDocument.docx ».

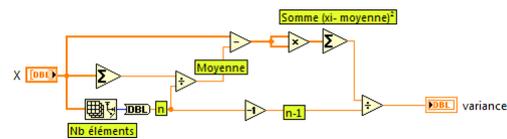
Un programme ou VI, développé dans l'environnement LabVIEW, se compose principalement de deux éléments étroitement associés et regroupés sous le même nom « nom\_application.vi » (l'extension .vi permet une reconnaissance immédiate par l'environnement LabVIEW). Ainsi nous avons :

La « face-avant » (*Front panel*) qui est l'interface utilisateur du programme au sens génie logiciel : définition des entrées/sorties de données accessibles par

l'utilisateur du programme (figure suivante). Cette notion sera reprise en détail dans le chapitre 3 (du livre).



Le « Diagramme » (diagram) qui est le programme de l'application ou code source. Il écrit sous la forme d'un diagramme flux de données en langage G : ensemble des icônes et des liaisons entre ces icônes utilisées (figure suivante). Cette partie de l'application est ce que l'on appelle le code source par opposition à l'interface utilisateur.



### 3.2 Les dix commandements du développeur LabVIEW

Le développeur LabVIEW devra suivre la synthèse des commandements suivants pour s'assurer que son code est conforme aux règles de développement.

1. Sur un seul écran ton code tiendra.
2. De gauche à droite tu coderas.
3. Les fils tu ne croiseras pas.
4. La gestion de l'erreur tu ne transgressas pas.
5. Le connecteur de type 4x4 toujours tu utiliseras.
6. Les contrôles, indicateurs et sous-VI judicieux tu nommeras.
7. Le flux de données tu respecteras.
8. Ton code toujours et immédiatement tu documenteras.
9. Des sous-VI tu feras.
10. La structure séquence empilée tu refuseras.

**Plus d'informations sur les règles de style à respecter avec LabVIEW dans le livre.**

## 4 Méthodologie pour trouver de l'aide sous LabVIEW

Lorsqu'un développeur débute, il est normal qu'il se pose de nombreuses questions et qu'il ait besoin d'aide. S'il est seul, comme c'est souvent le cas, il peut rapidement se trouver dans une impasse et perdre beaucoup de temps. Il est très important de connaître les bonnes méthodologies à appliquer pour trouver les bonnes réponses.

### 4.1 Recherche dans l'aide contextuelle

Sur une fonction de la palette, le moyen le plus simple est d'utiliser la fenêtre d'aide contextuelle : menu Aide, puis Visualiser l'aide ou Ctrl-H : si cette aide est active, il suffit de placer le curseur sur un élément pour avoir une aide simplifiée. Dans la fenêtre, il est possible d'accéder à une aide plus détaillée en cliquant sur le lien hypertexte.

### 4.2 Recherche dans l'aide LabVIEW

Pour vous procurer des informations sur les concepts de programmation LabVIEW, des instructions détaillées : sélectionnez Aide puis Rechercher dans l'Aide LabVIEW.

### 4.3 Recherche d'exemples

Pour développer une nouvelle fonction, vous pouvez modifier un VI d'exemple pour l'adapter à votre application : Sélectionnez Aide puis Recherche d'exemples pour parcourir des centaines de VI d'exemple (chapitre 4).

### 4.4 La communauté travaille pour vous

Internet et surtout le site de National Instruments, [www.ni.com](http://www.ni.com), regroupent des liens vers des cours et formations, solutions utilisateur (voir paragraphe 6.6), VI d'exemple, support technique, groupes d'utilisateurs et forum. Il est très riche en contenu et constitue une réelle source d'inspiration. À noter qu'il existe en langue française :

- deux forums de discussions pour NI LabVIEW et autres sujets, avec des experts qui répondent rapidement à toutes les questions. Les forums francophones sont d'une très grande qualité. Pour les trouver, le plus simple est de faire une recherche sur Internet avec les mots-clés : « Discussions au sujet de NI LabVIEW »

. J'interviens régulièrement sur le forum : [lien 57](#) ;

- une communauté très dynamique qui organise des rencontres développeurs pour le partage de techniques, d'informations et de convivialité ;
- les blogs des développeurs certifiés, par exemple blog du développeur certifié LabVIEW, Luc DESRUELLE, pour plus d'exemples : [lien 58](#).



Figure 3.10 - Pour trouver de l'aide, allez dans l'aide contextuelle, le site NI France, les deux forums francophones ainsi que les blogs.

### 4.5 Contacter un partenaire certifié du « Programme Alliance »

Pour une expertise, un besoin ponctuel ou pour sous-traiter un développement, nous vous conseillons de faire appel aux partenaires National Instruments du **Programme Alliance**. Ces partenaires ont reçu, après évaluation, l'agrément certifiant de leurs compétences en matière de programmation et de connaissances générales sur le logiciel LabVIEW. La liste est disponible sur le site de National Instruments. Vous aurez ainsi accès à des professionnels disposant des niveaux Certifié LabVIEW Développeur (CLD), voire Certifié LabVIEW Architecte (CLA). Ils seront les garants de la réussite de vos projets, avec un développement dans les règles de l'art. Pour les trouver, le plus simple est de faire une recherche sur Internet avec les mots-clés « partenaire alliance national instruments ».

*Moi, Luc DESRUELLE, l'auteur du livre LabVIEW et de ce tutoriel, serait ravi de vous apporter mon expertise en tant que Certifié LabVIEW Architecte (CLA), au travers de la société MESULOG qui est partenaire National Instruments. Mais aussi l'ensemble de mes confrères...*

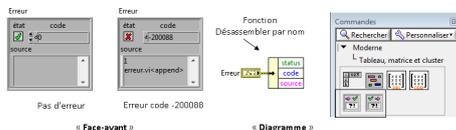
## 5 La Gestion des erreurs sous LabVIEW

### 5.1 Définition

La gestion de l'erreur n'est jamais une option, mais une obligation. Elle consiste à anticiper, détecter et résoudre les mises en garde et les erreurs. Sans cette gestion, le programme peut avoir un comportement incohérent, mais sans pouvoir en identifier la cause. Par exemple, sans gestion d'erreur, un programme pourrait réaliser une acquisition sur un instrument éteint, et sauvegarder sans problème les données sur un disque dur réseau qui n'existe pas. Un miracle ? Non un calvaire.

La structure de la donnée « erreur » au niveau de LabVIEW est un cluster formé des trois éléments suivants (figure 3.22) :

- **Etat** : un booléen qui est à l'état « Vrai » si une erreur s'est produite ;
- **Code**, un entier de 32 bits, qui contient le numéro de l'erreur ou « 0 » s'il n'y a pas d'erreur. Un code d'erreur différent de zéro avec un état « Faux » signale une mise en garde. Une mise en garde n'empêche pas le code de tourner, mais LabVIEW attire votre attention sur un point qu'il lui semble anormal.
- **Source** : une chaîne de caractères identifiant l'endroit où l'erreur s'est produite.

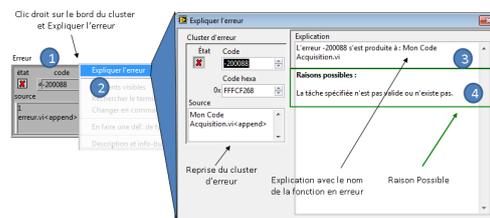


**Figure 3.22 - Type de représentation de la donnée « erreur » (cluster) et extraction des différents éléments avec la fonction Désassembler par nom, utilisée dans le « Diagramme ».**

### 5.2 Comprendre l'origine et la raison de l'erreur

En code source, pour résoudre le problème, il est possible de faire un clic droit sur le bord du cluster d'erreur et de sélectionner « Expliquer l'erreur » (figure 3.23). Une boîte de dialogue de gestion de l'erreur va s'ouvrir permettant :

- d'expliquer l'origine en visualisant le nom du VI qui a généré l'erreur, dans notre exemple « Mon Code Acquisition.vi » ;
- de proposer une ou des raisons possibles par rapport au code d'erreur, dans notre exemple LabVIEW propose la raison suivante : « La tâche spécifiée n'est pas valide ou n'existe pas », ce qui est le cas ; L'origine et la cause permettent en général de comprendre le problème et de le résoudre. Si cela n'est pas le cas, il faut utiliser les outils de débogage, comme les sondes et les points d'arrêt.



**Figure 3.23 - Comprendre l'origine de l'erreur et la raison pour corriger le problème.**

### 5.3 Gestion dans les VI

Tous les VI posséderont donc obligatoirement un « cluster d'entrée erreur » et un « cluster de sortie erreur ». Une exception peut être faite concernant les VI qui ne génèrent aucune erreur. Par exemple, un VI qui réalise le calcul «  $a^2+b^2$  ». Ce calcul est réalisé par des fonctions natives de LabVIEW qui ne génèrent pas d'erreur, le VI qui réalise ce code ne générera donc pas d'erreur. Cependant, même dans ce cas, il est intéressant d'ajouter à ce VI une entrée et une sortie « cluster d'erreur ». Cette entrée et cette sortie seront directement reliées entre elles. Ce « dispositif » permettra de positionner le VI dans le flux d'erreur global et donc de séquencer le code de la bonne façon.

 **Plus d'informations sur la gestion des erreurs avec LabVIEW dans le livre.**

## 6 Création d'un objet personnalisé : exemple d'un GIF animé

Pour d'autres exemples de personnalisation d'objets LabVIEW, voir le livre chapitre 3.

Pour un aspect plus ergonomique, il est possible de personnaliser les « objets graphiques » de la face-avant. Par exemple, il est possible d'importer des images, par exemple en utilisant un objet de type **menu déroulant d'images** (*picture Ring*). Il est

conseillé d'utiliser la fonction Édition ? Importer une image dans le presse-papier afin de ne pas dégrader l'image, puis de sélectionner Ajouter un élément après par un clic droit sur le contrôle. L'état à afficher est disponible via la valeur de l'indicateur (figure 3.39).

Il est également possible d'ajouter une image ani-

mée pour signaler le déroulement d'une action en cours sans avoir besoin de saisir du code (figure 3.40). Par exemple, pour afficher une mise à jour, nous allons insérer un GIF animé dans l'état « vrai » d'un bouton, et aucune image dans l'état « faux ».

Cela permettra de visualiser cette mise à jour en affichant l'état « vrai » du bouton. La fin de la mise à jour correspondra à l'état « faux » du même bouton.

Après avoir réalisé ou téléchargé une image libre de droits au format Gif animé, positionnez une commande de type Classique? Booleen? Bouton carré plat de la palette sur la « face-avant ». Sélectionnez Personnaliser la commande (Customize Control...) du menu Édition (Edit). La fenêtre de l'éditeur de commande apparaît. Sélectionnez dans le menu : Édition? Importer une image dans le presse-papier afin de ne pas dégrader l'image, et sélectionnez l'image animée. Faites un clic droit sur le contrôle et sélectionnez Importer une image du presse-papier? Vrai, nous avons modifié l'image du bouton pour l'état « Vrai ».

Figure 3.39 - Exemple d'une personnalisation simple d'un objet indicateur.

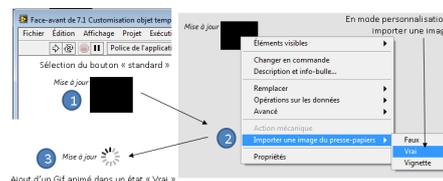


Figure 3.40 - Exemple d'une personnalisation des images d'un objet.

L'utilisation d'images et de couleurs permet de mieux respecter les critères subjectifs comme les conventions et la culture. Plus d'exemples d'objets personnalisés dans le livre et sur le blog une image pour éviter de faire du code : lien 59



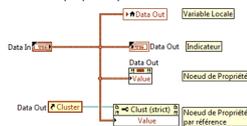
Programmation LabVIEW personnaliser un bouton en important une image gif, cliquez pour visualiser la vidéo : lien 60 .

## 7 Gestion avancée des données

### 7.1 Indicateur VS Locale VS Globale VS Nœud de propriété

Savez-vous répondre à la question (simple) suivante? (niveau CLAD...)

- Pour transmettre la donnée « Data In », faut-il mieux connecter le contrôle sur :
- A La variable locale de « Data Out »
  - B Le terminal de l'indicateur « Data Out »
  - C Le nœud de propriété implicite « value » de « Data Out »
  - D Le nœud de propriété par référence « value » de « Data Out »

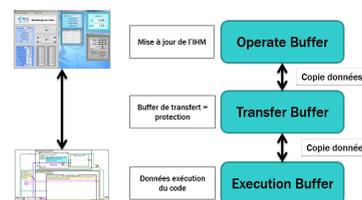


### Indicateur VS locale VS globale VS Nœud de propriété

L'explication n'est pas si simple.

Par définition les nœuds de propriété se synchronisent dans le thread « interface utilisateur » :

- seul système d'exécution non multithread, donc si occupé provoque des lenteurs ;
- la face-avant du VI sera chargée en mémoire. Or les commandes et indicateurs, sur une face-avant chargée en mémoire, ont leur propre copie des données. Il en résulte trois copies (buffer) des variables pour la mise à jour entre le diagramme et l'IHM.



Les commandes et indicateurs, sur une face-avant chargée en mémoire, ont leur propre copie des données.

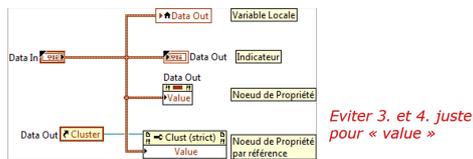
<p>□ Nœud de propriété</p> <ul style="list-style-type: none"> <li>• Copie donnée « Transfert Buffer »</li> <li>• Que Thread UI</li> <li>• Copie donnée « Operate Buffer »</li> </ul> <p>• Force IHM en mémoire</p> <p>• Race Condition</p>	V.S.	<p>□ Locale</p> <ul style="list-style-type: none"> <li>• Copie donnée « Transfert Buffer »</li> <li>• Tout les Threads</li> <li>• Copie donnée « Operate Buffer » seulement si IHM en mémoire</li> <li>• Race condition</li> </ul>
--	------	--



Mettre à jour une variable locale est « plus rapide », mais n'est pas forcément « la meilleure » solution.

La comparaison avec la variable globale engendre la même conclusion. Au final

1. L'indicateur « Data Out »
2. La variable locale (copie buffer)
3. Le nœud de propriété implicite (thread UI + copie buffer)
4. Le nœud de propriété par référence (référence)



**Conclusion ?**

Mais même si ces techniques sont simples à mettre en œuvre, **il est conseillé d'éviter l'utilisation des variables Locale - Globale - nœud de propriété**, car elles engendrent :

- copie de données ;
- pas de protection contre les « Race Conditions » ;
- pas de gestion d'erreur.

Les mises à jour des nœuds de propriétés sont réalisées via le thread de l'interface utilisateur, ce qui peut engendrer des ralentissements. Pour transmettre les données, **il est conseillé d'utiliser** :

- l'indicateur ;
- un fil de liaison ;
- des techniques spécifiques à LabVIEW comme la FGV, AE, DVR.

Pour des explications détaillées, vous référer au chapitre 3 du livre.

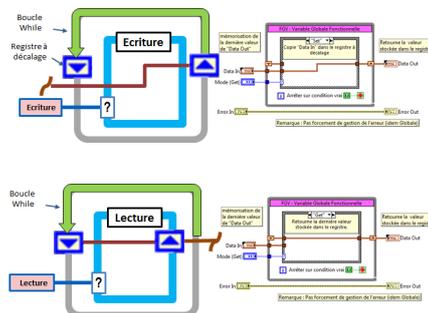
permet d'apporter des solutions aux problèmes des locales et globales que nous avons exposés au paragraphe précédent.

Pour reprendre la définition, une VGF (FGV) est composée :

- d'une boucle While, avec la condition d'arrêt qui est une constante « vraie » pour qu'elle ne s'exécute qu'une seule fois ;
- d'un registre à décalage non initialisé, qui mémorise la valeur précédente de la donnée ;
- d'une structure « condition », avec deux états « Écriture » et « Lecture » ;
- d'une commande de type énumération (ou énum), avec deux valeurs « Écriture » et « Lecture ».

**7.2.b Principe et bénéfice**

Une FGV est donc un registre à décalage non initialisé d'un VI non réentrant, qui permet de mémoriser une valeur dans le registre lors d'une commande « Écriture » (noté souvent « Set »), et de la retourner sur un « Lecture » (noté souvent « Get »), comme illustré à la figure 3.51.



*Figure 3.51 - Représentation symbolique et par le code du principe de la FGV, un registre à décalage non initialisé.*

**7.2 Variable Globale Fonctionnelle ou VGF (FGV)**

**7.2.a Histoire**

Ce n'est qu'à partir de la version 2 que l'équipe de développement du logiciel LabVIEW a défini que la valeur d'un registre à décalage non initialisé conserverait sa valeur précédente. L'autre option aurait été qu'elle reprenne sa valeur par défaut. À cette époque il n'y avait pas de globale « native » dans le logiciel. Cette technique s'est logiquement imposée comme la méthode pour partager des données dans une application sous le nom de « Globale LV2 », pour LabVIEW version 2, ou « Variable Globale Fonctionnelle (VGF) ». Mais son appellation la plus communément répandue est son nom anglais *Functional Global Variable* (FGV). Même avec l'arrivée des globales dans la version 3 de LabVIEW, cette méthode de développement a été conservée, car elle

Les bénéfices apportés par cette technique

- Une meilleure gestion des données en mémoire, car le stockage de l'information est unique, soit dans le registre à décalage du VI. Il supprime ainsi toutes les copies mémoire ou buffer, engendrées par la présence des globales et/ou locales. Cette méthode est d'un grand intérêt, surtout pour des structures de données complexes comme des tableaux de cluster.
- Possibilité d'ajouter une gestion de l'erreur, via le cluster d'erreur standard de LabVIEW et une structure condition. Il est ainsi possible d'éviter toute opération sur la FGV si une erreur est présente à son entrée. Il est à noter que cela est optionnel, mais conseillé.



Plus d'informations sur la gestion des données, notamment FGV, AE, DVR, programmation objet (OOP) sous LabVIEW dans le livre et le blog de Luc Desruelle.

Retrouvez la suite de l'article de **Luc Desruelle**, **Francis Cottet** et **Michel Pinard** en ligne : [lien 61](#)

# NoSQL



## Les dernières news NoSQL

# Les meilleurs cours et tutoriels pour apprendre les bases de données NoSQL

La rubrique NoSQL vient d'ajouter une nouvelle page dédiée aux tutoriels. Nous avons décomposé en cinq grandes familles :

- Généralités
- Tutoriels MongoDB
- Tutoriels Apache Cassandra
- Tutoriels Neo4j
- Autres

Vous y trouverez des tutoriels sur Apache Cassandra, MongoDB, Neo4j, Mahout et autres :

- Apprendre à modéliser avec Apache Cassandra (lien 62)
- Comparatif Apache Flink et Apache Spark : redondance ? (lien 63)
- Introduction à Neo4j, une base de données orientée graphe - Partie 1 : Concept, installation, première prise en main et écosystème (lien 64)

- Introduction à la base de données NoSQL MongoDB (lien 65)
- Tutoriel sur le développement Java avec Cassandra 2 (lien 66)
- Formation MongoDB pour les développeurs Java (lien 67)
- Introduction à Apache Mahout pour la mise en place d'outils de recommandation (lien 68)
- Modélisation d'un schéma d'une base de données NoSQL orientée document (lien 69)

Vous aussi, vous pouvez contribuer à la rubrique NoSQL, nous proposer des ressources à publier sur Developpez.com ou contribuer à la mise en place des FAQ NoSQL. Vous pouvez également aider des débutants sur le forum d'entraide NoSQL.

- Page cours NoSQL : lien 70
- FAQ Apache Cassandra : lien 71
- Forum d'entraide NoSQL : lien 72

Commentez la news de **Mickael Baron** et **Francis Walter** en ligne : [lien 73](#)

# Solutions d'Entreprise

## Les dernières news Solutions d'Entreprise

# Les meilleurs cours et tutoriels sur les processus métier

L'équipe de la rédaction a mis en place une page cours qui rassemble tous les cours qui portent sur les processus métier ([lien 74](#)) qui ont été publiés sur nos rubriques. La page contient pour l'instant sept cours et tutoriels qui sont répartis comme suit :

- Généralité
- Notation BPMN
- Outils de modélisation BPM

Sur cette page cours, vous découvrirez les meilleurs cours, tutoriels, articles et billets de blogs

sur les processus métier et le workflow. Apprenez ce que c'est que le BPM et comment modéliser les processus avec la notation BPMN. Vous découvrirez également sur cette page des tutoriels sur l'utilisation des outils tels que Bonita BPM, JBoss JBPM et bien d'autres à venir sur W4 Modeler.

- Page cours BPM : [lien 75](#)
- Forum BPM : [lien 76](#)
- Page livres BPM : [lien 77](#)

Commentez la news de **Francis Walter** en ligne : [lien 78](#)

# Libres & Open Source



## Les blogs Open Source

# Tour d'horizon des distributions éducatives

Que nous soyons parents, éducateurs ou enseignants, nous sommes inévitablement conduits à nous poser le problème de la formation des enfants à l'informatique et de l'usage de l'ordinateur à des fins éducatives. Il existe de nombreuses offres logicielles commerciales, ou gratuites, mais nous savons peut-être moins qu'il existe des distributions libres entièrement destinées à l'éducation, structurées sur un système d'exploitation intégrant de nombreux outils éducatifs.

Le cahier des charges pour un PC éducatif n'est pas le même que pour un PC classique. Il convient en effet de s'assurer d'une part d'un minimum de sécurité (personne ne souhaite qu'un enfant ou un élève se retrouve sur un site inapproprié!), d'autre part que les logiciels proposés aient un contenu et une présentation adaptés aux enfants. C'est entre autres à ce type de problématique que tentent de répondre ces distributions éducatives. Notons toutefois que les protections n'étant pas toujours infaillibles il reste plus prudent de ne pas laisser un enfant naviguer sur Internet sans une présence adulte. De plus, ces distributions généralement peu gourmandes en ressources nous permettront de donner une seconde vie à d'anciens PC.

Cet article se propose de faire un tour d'horizon de quelques solutions existantes et de vous guider dans vos choix en fonction de vos besoins. Les solutions retenues ici peuvent aussi bien être utilisées dans un cadre familial que dans un cadre scolaire.

## 1 Abulédu

Très proche de la distribution ASRI Edu, mais moins riche en logiciels éducatifs préinstallés, Abulédu, basée sur Ubuntu, est proposée en live DVD ou USB que vous pouvez aussi installer sur disque dur : lien 79



L'environnement nécessite un minimum de 512 Mo pour fonctionner. Vous devrez également disposer d'un lecteur de DVD, ou bien d'un PC capable de démarrer sur une clé USB. Sa force réside dans le fait que cette distribution est à la base un serveur et un espace numérique de travail pour les réseaux des établissements scolaires. Même si elle est prioritairement destinée aux établissements scolaires, elle pourra également trouver sa place à la maison.

## 2 ASRI Edu

ASRI Education est une association visant à faciliter l'accès de l'outil informatique aux enfants en privilégiant les suites logicielles libres et gratuites et la revalorisation de matériels informatiques anciens.

L'association propose sur son site <http://asri-education.org> une riche logithèque, ainsi qu'un environnement libre complet basé sur Puppy Linux incluant ces logiciels. Avant de télécharger une des distributions

proposées, vous devrez choisir la plus adaptée à votre matériel en suivant le comparatif proposé sur cette page : [lien 80](#)



Elle propose un environnement très complet et riche en logiciels ludo-éducatifs, adaptables à différents âges jusqu'à une douzaine d'années par un simple changement de session. Une session « Profs » permettant la configuration de l'ordinateur et de certains logiciels est également disponible. Certaines versions sont en mesure de fonctionner avec de très vieux PC.

La distribution est proposée en live CD ou USB, c'est-à-dire que vous pouvez la démarrer à partir du lecteur de CD ou d'un port USB. Vous pouvez également l'installer sur le PC.

### 3 DoudouLinux

DoudouLinux est une distribution uniquement en live CD non installable destinée à faciliter l'usage de l'informatique aux enfants de 2 à 12 ans dans un espace sécurisé. Elle est donc facilement utilisable partout, sans connaissances techniques, et sans nécessiter de surveillance parentale. Elle est en mesure de fonctionner sur de vieux PC. Son seul inconvénient est la lenteur de démarrage liée à son fonctionnement à partir du CD. Pour pallier cet inconvénient vous pourrez, si le PC permet le démarrage sur un port USB, créer une clé live USB, ou bien encore, et bien que ce ne soit pas sa vocation première, l'installer sur le disque dur à certaines conditions ; voir ici : [lien 81](#)



Adresse de la page de téléchargement : [lien 82](#)

### 4 École numérique pour tous (ENPT)

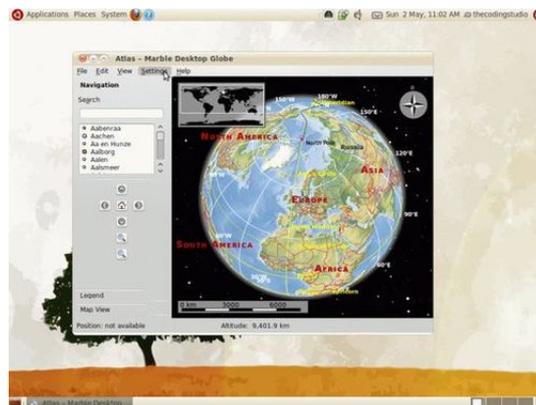
École numérique pour tous est une distribution destinée à l'éducation basée sur Ubuntu et proposée en live DVD ou USB. L'installation sur le disque dur est possible, mais uniquement à partir du démarrage en live. Elle ne propose pas d'options de démarrage ou d'installation au lancement du DVD ou de la clé USB aussi pourra-t-elle poser problème en cas de carte graphique très ancienne ou non reconnue.



Elle est adaptée pour les écoles de l'enseignement primaire et propose un accès pour les élèves et les enseignants. Sur l'écran d'accueil, un menu en mosaïque permet de choisir parmi des activités classées par thèmes. Voici l'adresse de la page sur laquelle vous en trouverez une description détaillée ainsi que le lien de téléchargement : [lien 83](#)

## 5 Edubuntu

Edubuntu est une version d'Ubuntu incluant de nombreux programmes éducatifs de la maternelle à l'université. C'est un live DVD ou USB que vous pourrez donc soit tester sans installation, soit installer sur disque dur. À défaut de disposer d'un lecteur de DVD ou de pouvoir démarrer à partir d'un port USB, vous pourrez procéder à l'installation d'Ubuntu à partir d'un live CD, puis reconstituer Edubuntu en installant les paquets Edubuntu desktop.



Cette distribution nécessite un PC suffisamment puissant, donc pas trop ancien. Pas d'environnements préadaptés selon les âges : les logiciels sont proposés en vrac dans la liste des applications du tableau de bord, sans organisation par thèmes ou âges. Vous devrez donc personnaliser le bureau avec des raccourcis vers les applications utiles en fonction de l'âge de vos enfants ou élèves. Si vous souhaitez disposer d'une liste des logiciels inclus classés selon divers critères vous pourrez vous rendre sur le portail éducation d'Edubuntu à l'adresse [lien 84](#)

Sur ce lien vous disposerez d'informations complètes : [lien 85](#), et vous pourrez la télécharger ici : [lien 86](#)

## 6 Freeduc

Freeduc est une distribution Linux sur CD, non installable, basée sur Knoppix et élaborée par l'organisation Ofset. Plusieurs versions existent, adaptées à différents niveaux de l'enseignement ou spécialisées dans un domaine. Ces distributions, assez anciennes maintenant, conviendront aisément à de très vieux PC. Les logiciels sont organisés par thèmes à partir de la barre des tâches du bureau.



Vous disposerez d'informations complètes et une vue claire des différentes versions ici : [lien 87](#)

Vous pourrez télécharger les différentes versions ici : [lien 88](#)

## 7 PrimTux

PrimTux est une distribution éducative complète développée en coopération avec l'association ASRI Education qui édite la distribution présentée précédemment. Elle en respecte la philosophie : des applications éducatives organisées par tranches d'âges auxquelles on accède par sessions séparées.



PrimTux se distingue de ASRI Edu notamment sur les points suivants :

- elle est construite à partir d'un noyau Debian (ASRI Edu est construite sur une Puppy Linux) et dispose ainsi d'une logithèque plus riche ;
- elle est plus récente, mais reste tout de même très légère. Elle passera peut-être moins facilement sur de très vieux PC. Le choix dépendra donc avant tout du matériel informatique dont vous disposez ;
- elle comporte davantage de logiciels et de ressources ;
- si les principes de ASRI Edu sont respectés, l'utilisation d'environnements de bureaux différents en modifie toutefois l'usage pratique.

PrimTux est livrée sous forme de live DVD installable d'une taille d'environ 3 Go. Vous pouvez bien entendu construire une clé USB bootable avec l'image fournie en utilisant un utilitaire comme Linux Live USB Creator par exemple : [lien 89](#).

Sur le bureau une fenêtre à onglets, le HandyMenu-PrimTux, donne accès aux applications éducatives classées par thèmes. L'adaptation aux âges de chaque session ne concerne pas seulement le choix des applications. C'est tout l'environnement qui est repensé : accès aux applications, menus... Jusqu'au logiciel de traitement de texte LibreOffice dont l'interface s'adapte aux plus jeunes enfants dès l'âge maternel, offre des fonctionnalités de facilitation de la lecture (coloriage syllabique par exemple)...

Dans la session Prof, le navigateur Internet (SeaMonkey) propose dans ses marque-pages une foultitude de liens vers des ressources éducatives classées par thèmes.

Vous trouverez une présentation détaillée de PrimTux en suivant ce lien : [lien 90](#) Adresse du site officiel sur lequel se trouvent les liens de téléchargement ainsi que de la documentation et des tutoriels : [lien 91](#) Le site dispose d'un forum sur lequel vous pourrez poser vos questions, faire remonter d'éventuels problèmes, proposer des améliorations : [lien 92](#)

## 8 ScolBuntu

ScolBuntu reprend le bureau et l'organisation d'École numérique pour tous (ENPT) sur la base d'une distribution plus légère en s'appuyant sur l'environnement de bureau Xfce. Elle présente l'avantage d'être utilisable sur des PC plus âgés, adaptation au vieux matériel favorisée par l'existence de deux versions dont l'une est construite sur un noyau plus ancien d'Ubuntu.



Elle est proposée en Live DVD ou USB installable, et contrairement à ENPT, démarre sur un menu qui permet de la lancer en mode vidéo de base (idéal pour des machines avec des cartes vidéo très anciennes ou exotiques), ou de l'installer directement sur disque dur sans passer par une session en live.

Vous en trouverez une présentation détaillée ainsi qu'une documentation et les liens de téléchargement à cette adresse : [lien 93](#)

## 9

Sur cette page, vous disposerez d'un tableau comparatif de certaines de ces distributions, avec des informations techniques (taille de l'image, date de sortie, environnement de bureau proposé) : [lien 94](#)

### 9.1 Si vous êtes pressé...

Si vous êtes pressé et souhaitez être rapidement orienté vers une solution vous convenant, je dirais que :

- si vous voulez pour une école maternelle ou primaire une solution globale pour les enseignants et les élèves, avec un accès à un serveur local et un tableau numérique, orientez-vous prioritairement vers Abuledu, École numérique pour tous (ENPT) ou Scolbuntu ;
- si vous recherchez des logiciels éducatifs pour des élèves de niveau secondaire ou supérieur, orientez-vous prioritairement vers Edubuntu ou Freeduc Science ;
- si vous voulez une solution plus orientée vers les logiciels éducatifs pour une classe de niveau maternel ou primaire, ou pour la maison, orientez-vous prioritairement vers PrimTux, ASRI Edu ou Freeduc École si votre machine est trop ancienne pour PrimTux, Doudoulinux si votre priorité est la sécurité de l'ordinateur et de l'enfant (accès Internet impossible).

Cet avis n'est toutefois pas absolu : ces distributions étant des systèmes d'exploitation à part entière, elles peuvent généralement être adaptées selon l'usage par l'ajout de logiciels et d'autres paramétrages à condition d'être installées sur le disque dur (sauf pour Doudoulinux).

### 9.2 Sources de l'article :

- Un PC éducatif sur de-bric-et-de-broc.fr : [lien 95](#)
- Les sites officiels des différentes distributions dont les liens sont donnés dans le corps de l'article avec chaque distribution.

### 9.3 Sources complémentaires :

- Distributions pour l'éducation sur Léa-Linux : [lien 96](#)
- Distributions scolaires sur Framasoft : [lien 97](#)

Retrouvez le blog de **Philippe Dpt35** en ligne : [lien 98](#)

# Liste des liens

## Page 2

lien 1 : ... <http://www.cmake.org/download/>

## Page 4

lien 2 : ... <http://www.cmake.org/cmake/help/v3.0/manual/cmake-language.7.html#syntax>

lien 3 : ... [http://www.cmake.org/cmake/help/v3.0/command/cmake\\_minimum\\_required.html](http://www.cmake.org/cmake/help/v3.0/command/cmake_minimum_required.html)

lien 4 : ... <http://www.cmake.org/cmake/help/v3.0/command/project.html>

lien 5 : ... [http://www.cmake.org/cmake/help/v3.0/command/target\\_include\\_directories.html](http://www.cmake.org/cmake/help/v3.0/command/target_include_directories.html)

lien 6 : ... [http://www.cmake.org/cmake/help/v3.0/command/target\\_link\\_libraries.html](http://www.cmake.org/cmake/help/v3.0/command/target_link_libraries.html)

lien 7 : ... [http://www.cmake.org/cmake/help/v3.0/command/add\\_executable.html](http://www.cmake.org/cmake/help/v3.0/command/add_executable.html)

lien 8 : ... [http://www.cmake.org/cmake/help/v3.0/command/add\\_library.html](http://www.cmake.org/cmake/help/v3.0/command/add_library.html)

lien 9 : ... <http://alexandre-laurent.developpez.com/tutoriels/cmake/fichiers/1-projet-simple.zip>

lien 10 : ... <http://alexandre-laurent.developpez.com/tutoriels/cmake/fichiers/2-projet-bib.zip>

## Page 5

lien 11 : ... [http://www.cmake.org/cmake/help/v3.0/command/add\\_subdirectory.html](http://www.cmake.org/cmake/help/v3.0/command/add_subdirectory.html)

lien 12 : ... [http://www.cmake.org/cmake/help/v3.0/command/target\\_include\\_directories.html](http://www.cmake.org/cmake/help/v3.0/command/target_include_directories.html)

lien 13 : ... <http://alexandre-laurent.developpez.com/tutoriels/cmake/fichiers/3-projet-ext-lib.zip>

lien 14 : ... [http://www.cmake.org/cmake/help/v3.0/command/find\\_package.html](http://www.cmake.org/cmake/help/v3.0/command/find_package.html)

## Page 6

lien 15 : ... [http://www.cmake.org/cmake/help/v3.1/command/target\\_compile\\_features.html](http://www.cmake.org/cmake/help/v3.1/command/target_compile_features.html)

lien 16 : ... [http://www.cmake.org/cmake/help/v3.1/prop\\_gbl/CMAKE\\_CXX\\_KNOWN\\_FEATURES.html](http://www.cmake.org/cmake/help/v3.1/prop_gbl/CMAKE_CXX_KNOWN_FEATURES.html)

lien 17 : ... <http://www.cmake.org/cmake/help/v3.0/command/install.html>

lien 18 : ... <http://www.cmake.org/cmake/help/v3.0/command/project.html>

lien 19 : ... [http://www.cmake.org/cmake/help/v3.0/command/configure\\_file.html](http://www.cmake.org/cmake/help/v3.0/command/configure_file.html)

lien 20 : ... [http://www.cmake.org/cmake/help/v3.0/command/find\\_package.html](http://www.cmake.org/cmake/help/v3.0/command/find_package.html)

## Page 7

lien 21 : ... [http://www.cmake.org/cmake/help/v3.0/command/add\\_executable.html](http://www.cmake.org/cmake/help/v3.0/command/add_executable.html)

lien 22 : ... [http://www.cmake.org/cmake/help/v3.0/command/add\\_executable.html](http://www.cmake.org/cmake/help/v3.0/command/add_executable.html)

## Page 8

lien 23 : ... <http://www.cmake.org/cmake/help/v3.0/command/install.html>

lien 24 : ... <http://nsis.sourceforge.net/Download/>

lien 25 : ... <http://www.cmake.org/Wiki/CMake:CPackConfiguration>

lien 26 : ... <http://alexandre-laurent.developpez.com/tutoriels/cmake/fichiers/4-projet-bib-cpack.zip>

lien 27 : ... <http://alexandre-laurent.developpez.com/tutoriels/cmake/>

## Page 19

lien 28 : ... <http://claudeleloup.developpez.com/tutoriels/access/menus7jours/Menus7Jours.mdb>

lien 29 : ... <http://claudeleloup.developpez.com/tutoriels/access/menus7jours/>

## Page 20

lien 30 : ... <http://www.atozed.com/index.EN.aspx>

lien 31 : ... <http://www.atozed.com/IntraWeb/Download/index.EN.aspx>

lien 32 : ... <http://app.atozed.com/>

## Page 21

lien 33 : ... <http://downloads.atozed.com/intraweb/IWBundledRemovalTool.zip>

## Page 25

lien 34 : ... <http://dany-leblanc.developpez.com/tutoriels/delphi/intraweb/>

## Page 26

lien 35 : ... <https://pypi.python.org/pypi>

lien 36 : ... <http://deussyss.developpez.com/tutoriels/Python/python-en-bref/#LIV-K-2-e>

Page 27

lien 37 : ... <https://pip.pypa.io/en/stable/installing.html>

Page 29

lien 38 : ... [https://pip.pypa.io/en/latest/reference/pip\\_install.html#vcs-support](https://pip.pypa.io/en/latest/reference/pip_install.html#vcs-support)

lien 39 : ... <https://www.python.org/dev/peps/pep-0427/>

lien 40 : ... <http://pythonwheels.com/>

lien 41 : ... [https://pythonhosted.org/an\\_example\\_pypi\\_project/setuptools.html](https://pythonhosted.org/an_example_pypi_project/setuptools.html)

lien 42 : ... <https://pythonhosted.org/setuptools/setuptools.html>

lien 43 : ... <https://docs.python.org/2/distutils/apiref.html?highlight=bdist>

Page 30

lien 44 : ... [https://pypi.python.org/pypi?%3Aaction=list\\_classifiers](https://pypi.python.org/pypi?%3Aaction=list_classifiers)

Page 32

lien 45 : ... <http://alexis.notmyidea.org/distutils2/setupcfg.html>

Page 35

lien 46 : ... <https://pypi.python.org/pypi/twine>

lien 47 : ... <https://packaging.python.org/en/latest/distributing.html#uploading-your-project-to-pypi>

Page 36

lien 48 : ... <https://pypi.python.org/pypi/>

lien 49 : ... <http://doc.devpi.net/latest/>

lien 50 : ... <http://doc.devpi.net/latest/userman/index.html>

Page 37

lien 51 : ... [http://doc.devpi.net/latest/userman/devpi\\_commands.html#devpi-command-reference-server](http://doc.devpi.net/latest/userman/devpi_commands.html#devpi-command-reference-server)

lien 52 : ... <http://doc.devpi.net/latest/web.html#themes>

Page 40

lien 53 : ... <http://deussyss.developpez.com/tutoriels/Python/SphinxDoc/>

Page 42

lien 54 : ... [http://deussyss.developpez.com/tutoriels/Python/packaging\\_pypi/](http://deussyss.developpez.com/tutoriels/Python/packaging_pypi/)

Page 43

lien 55 : ... <http://labview.developpez.com/livres/index/?page=Les-meilleurs-livres-LabVIEW#L2100724339>

lien 56 : ... <http://www.amazon.fr/exec/obidos/ASIN/2100724339/wwwdeveloppec-21>

Page 46

lien 57 : ... <http://forums.ni.com/t5/Discussions-au-sujet-de-NI/bd-p/4171>

lien 58 : ... [https://decibel.ni.com/content/blogs/Luc\\_Desruelle](https://decibel.ni.com/content/blogs/Luc_Desruelle)

Page 48

lien 59 : ... [https://decibel.ni.com/content/blogs/Luc\\_Desruelle/2014/03/23/un-gif-animé-pour-éviter-de-faire-du-code](https://decibel.ni.com/content/blogs/Luc_Desruelle/2014/03/23/un-gif-animé-pour-éviter-de-faire-du-code)

lien 60 : ... <https://vimeo.com/104782440>

Page 50

lien 61 : ... <http://luc-desruelle.developpez.com/tutoriels/labview/extrait-livre-programmation-application/>

Page 51

lien 62 : ... <http://ippon.developpez.com/tutoriels/big-data/cassandra/modelisation-cassandra/>

lien 63 : ... <http://ippon.developpez.com/tutoriels/big-data/apache-flink-park/redondance/>

lien 64 : ... <http://logisima.developpez.com/tutoriel/nosql/neo4j/introduction-neo4j/>

lien 65 : ... <http://harry-wanki.developpez.com/tutoriels/mongodb/debuter-mongodb-introduction-base-donnees-nosql/>

lien 66 : ... <http://zenika.developpez.com/tutoriels/nosql/cassandra-developpement-java/>

lien 67 : ... <http://arolla.developpez.com/tutoriels/nosql/formation-mongodb-developpeurs-java/>

lien 68 : ... <http://soat.developpez.com/tutoriels/nosql/introduction-mahout-devoxfr2014/>

- lien 69 : ... <http://paloit.developpez.com/tutoriels/nosql/modelisation-schema-base-donnees-orientee-document/>
- lien 70 : ... <http://nosql.developpez.com/cours/>
- lien 71 : ... <http://nosql.developpez.com/faq/cassandra>
- lien 72 : ... <http://www.developpez.net/forums/f1743/bases-donnees/nosql/>
- lien 73 : ... <http://www.developpez.net/forums/d1388617/bases-donnees/nosql/meilleurs-cours-tutoriels-apprendre-bases-donnees-nosql/>

#### Page 52

- lien 74 : ... <http://bpm.developpez.com/cours/>
- lien 75 : ... <http://bpm.developpez.com/cours/>
- lien 76 : ... <http://www.developpez.net/forums/f2110/logiciels/solutions-d-entreprise/bpm/>
- lien 77 : ... <http://bpm.developpez.com/livres/>
- lien 78 : ... <http://www.developpez.net/forums/d1549684/logiciels/solutions-d-entreprise/bpm/meilleurs-cours-tutoriels-processus-metier/>

#### Page 53

- lien 79 : ... <http://www.abuledu.org/abuledu/monoposte>

#### Page 54

- lien 80 : ... <http://asri-education.org/asriedu-telechargeriso>
- lien 81 : ... <http://www.doudoulinux.org/web/francais/documentation/outils-avances/article/installer-doudoulinux.html>
- lien 82 : ... <http://www.doudoulinux.org/web/francais/index.html>

#### Page 55

- lien 83 : ... <http://www.imaugis.com/la-distribution-ecole-numerique-pour-tous>
- lien 84 : ... <http://doc.ubuntu-fr.org/education>
- lien 85 : ... <http://doc.ubuntu-fr.org/edubuntu>
- lien 86 : ... <http://www.edubuntu-fr.org/telechargement>

#### Page 56

- lien 87 : ... <http://fr.wikipedia.org/wiki/Freeduc>
- lien 88 : ... <http://sourceforge.net/projects/ofset/files/freeduc-cd/>
- lien 89 : ... <http://linux.developpez.com/telecharger/detail/id/3934/Linux-Live-USB-Creator>
- lien 90 : ... <http://printux.developpez.com/tutoriels/demarrage-rapide/>
- lien 91 : ... <http://printux.fr/>
- lien 92 : ... <http://forum.printux.fr/>

#### Page 57

- lien 93 : ... <http://aft-rn.net/spip3/spip.php?article180>
- lien 94 : ... <http://socialcompare.com/fr/comparison/distributions-linux-francophones-pour-les-ecoliers>
- lien 95 : ... <http://www.de-bric-et-de-broc.fr/que-faire-d-un-vieux-pc/un-pc-educatif>
- lien 96 : ... [http://lea-linux.org/documentations/Logith%C3%A8que:Distinctions\\_pour\\_l\\_education](http://lea-linux.org/documentations/Logith%C3%A8que:Distinctions_pour_l_education)
- lien 97 : ... <http://framsoft.net/rubrique213.html>
- lien 98 : ... <http://www.developpez.net/forums/blogs/928180-philippe-dpt35/b883/tour-d-horizon-distributions-educatives/>