

INSTALLATION ET CONFIGURATION D'UN SERVEUR WEB

(Apache + Php + Mysql)

PAR

Maisse Sébastien



1. Introduction.....	
2. Installation et configuration d'Apache.....	
3. Installation et configuration de Php 4.....	
4. Installation et configuration de Mysql.....	
5. Installation et configuration de Phpmyadmin.....	
6. Introduction à la sécurité d'un serveur.....	
7. Ressources et Conclusion.....	

INTRODUCTION

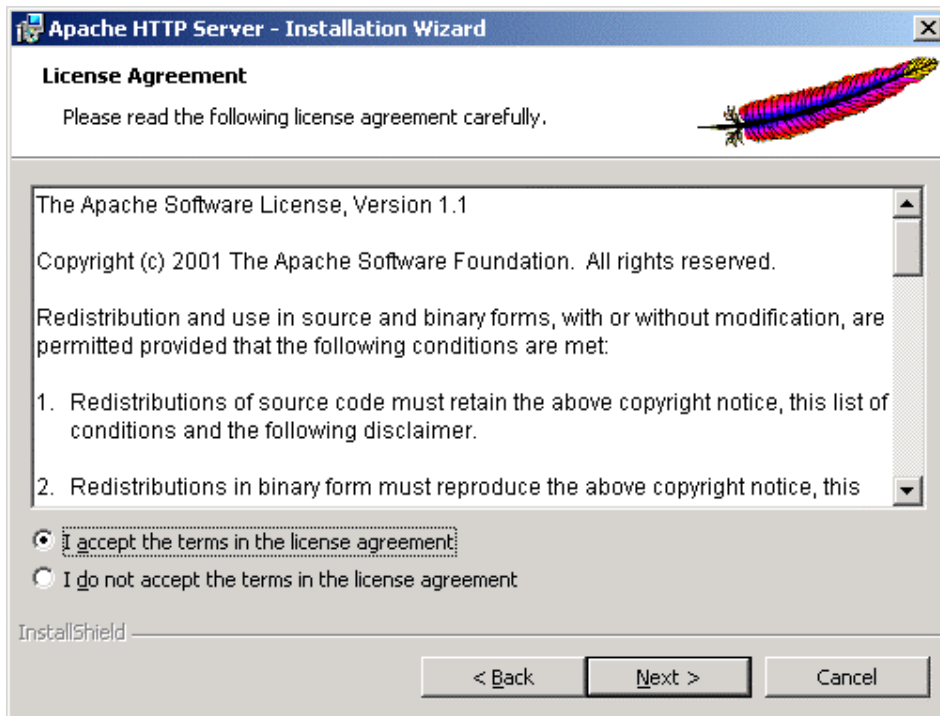
Vous souhaitez créer votre propre serveur Web, parce que vous avez une connexion haut débit (adsl par exemple) qui vous permet d'accueillir et d'héberger vos pages chez vous sans passer par un hébergement. Cela est tout ça fait envisageable, il vous suffit juste d'installer de quoi transformer votre ordinateur en petit serveur Web perso :) Ce que je vous propose de faire grâce à ce petit tutorial qui vous explique les différentes étapes de l'installation pour devenir votre propre hébergeur. **ATTENTION** : le tutorial concerne l'installation sur un ordinateur qui est pourvu de Windows 2000 Professionnel (seul système à ma disposition à l'heure où je rédige ce tutorial), sachez que l'installation diffère peu d'un système à l'autre.

INSTALLATION ET CONFIGURATION DE APACHE 1.3

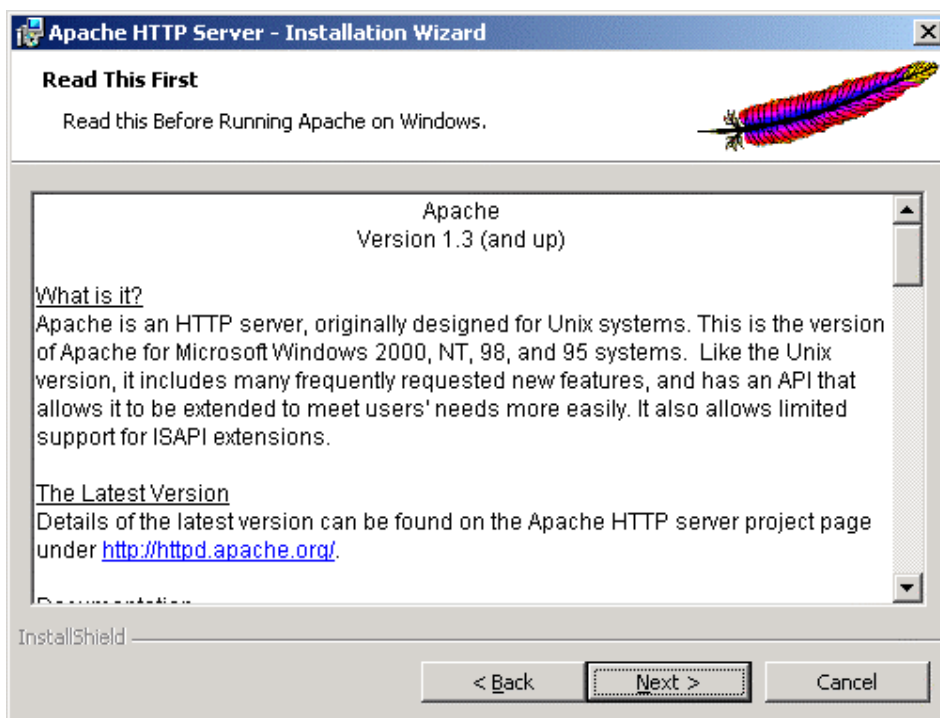
Tout d'abord, nous devons en premier lieu télécharger ce qui va nous permettre d'installer notre serveur Web, pour cela je vous conseil de vous rendre sur le site officiel <http://www.apache.org> ou sur le site <http://www.geldenduys.com>, de là, il vous faut vous rendre dans la rubrique **Dernières Versions** et télécharger **Apache 1.3.27** (pour information, il existe deux versions de Apache, celle que l'on va installer ainsi que la version **2.0.X**, cette dernière présentant, pour le moment, une incompatibilité avec le **php**) c'est la version qui nous intéresse pour mettre en place notre serveur web. Une fois le téléchargement terminé (taille ~ 5,3 Mo), nous exécutons le programme d'installation de apache. Ceci fait, vous devriez avoir l'écran ci-dessous qui s'affiche lors du lancement du programme :



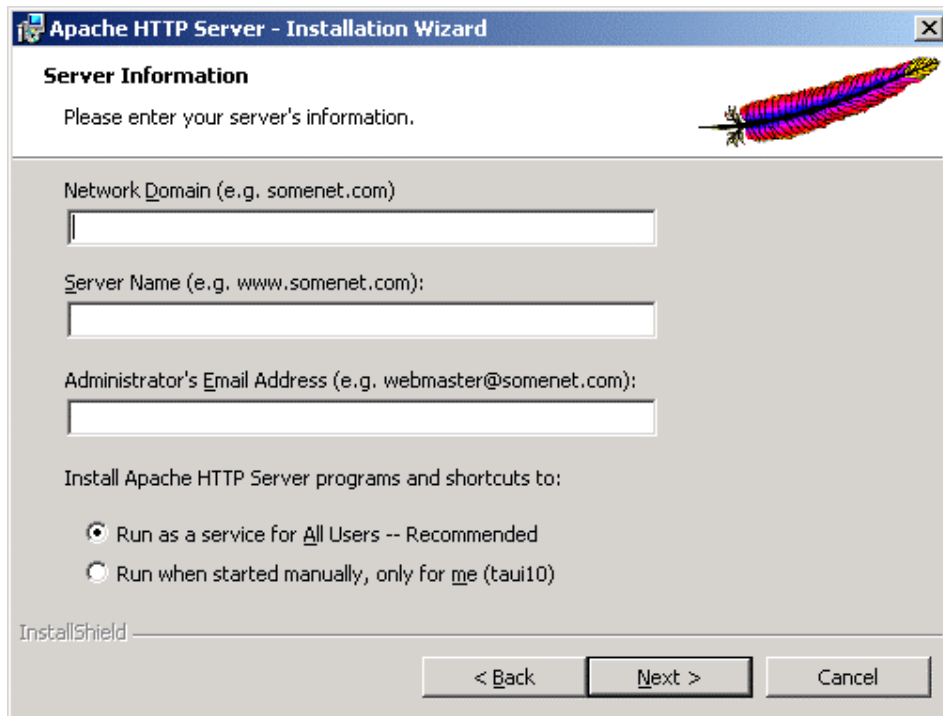
Cliquer sur le bouton « *Next* », là, l'écran suivant s'affiche :



Cliquer sur « *I accept the terms in the license agreement* » et sur le bouton « *Next* »



Un petit texte d'introduction, cliquer sur le bouton « *Next* » pour passer à la suite...



Là, un 1^{er} écran de configuration nous invite à rentrer les informations suivantes :

Network Domain :

Le nom du domaine de votre ordinateur (ici ce sera localhost).

Server Name :

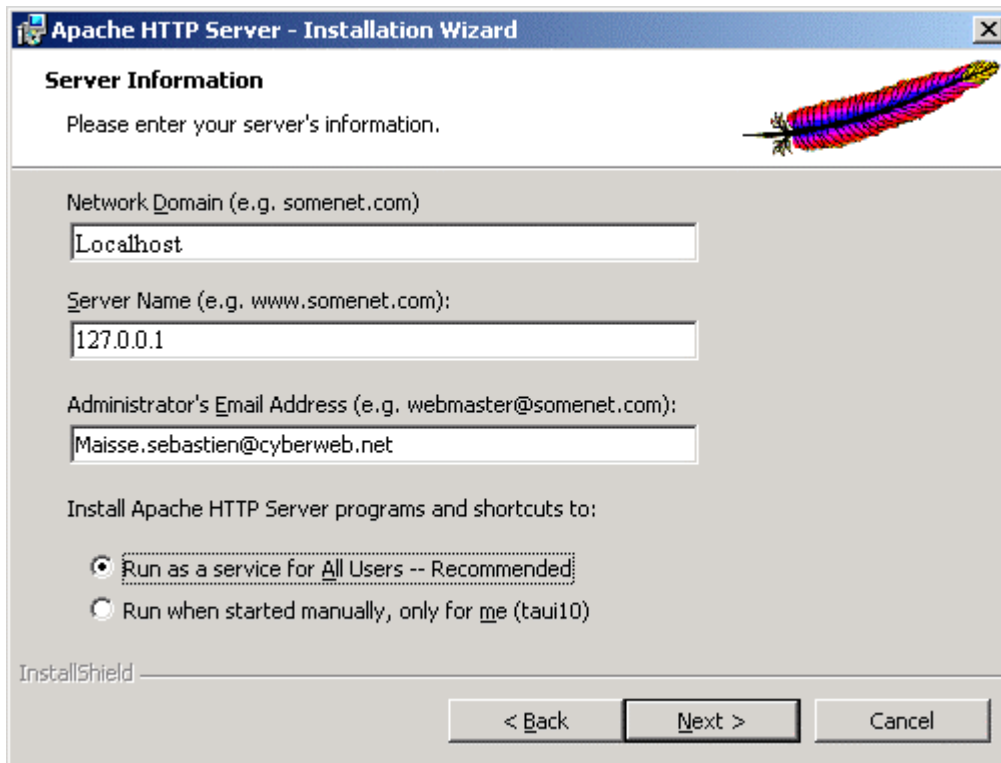
Le nom que l'on va donner à notre serveur.

Administrator's Email Address :

L'adresse email de l'administrateur.

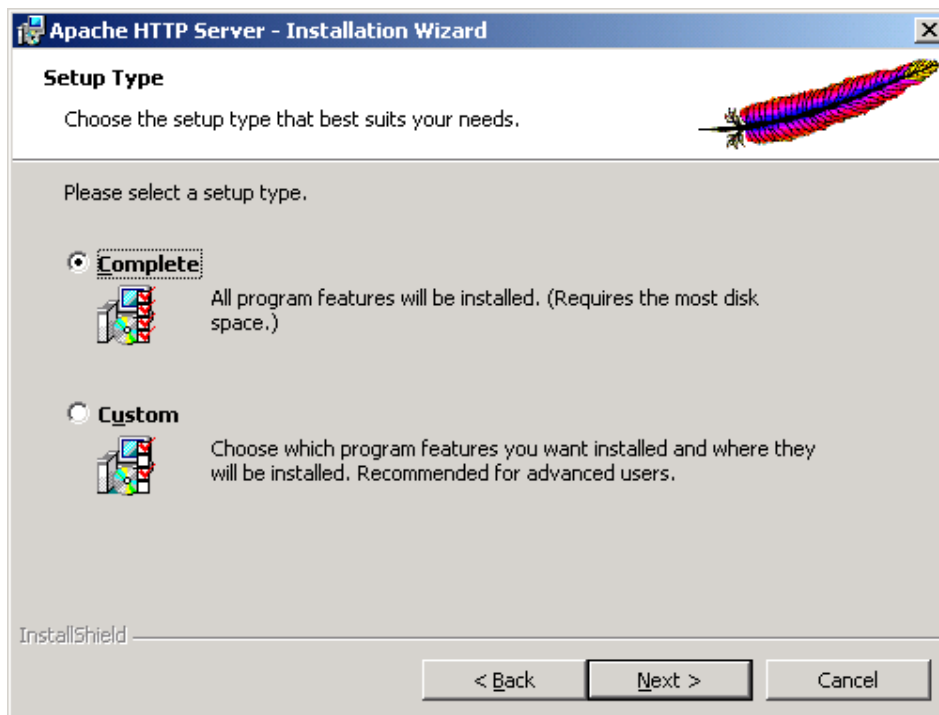
Install Apache http Server programs and shorcuts to :

Cette option nous permet soit de faire l'installation pour tout les utilisateurs (**Run as a service for all Users**) ou juste pour l'utilisateur en cours de session (**Run when started manually, only for me**, dans la cas présent **taui10**) sur l'ordinateur.



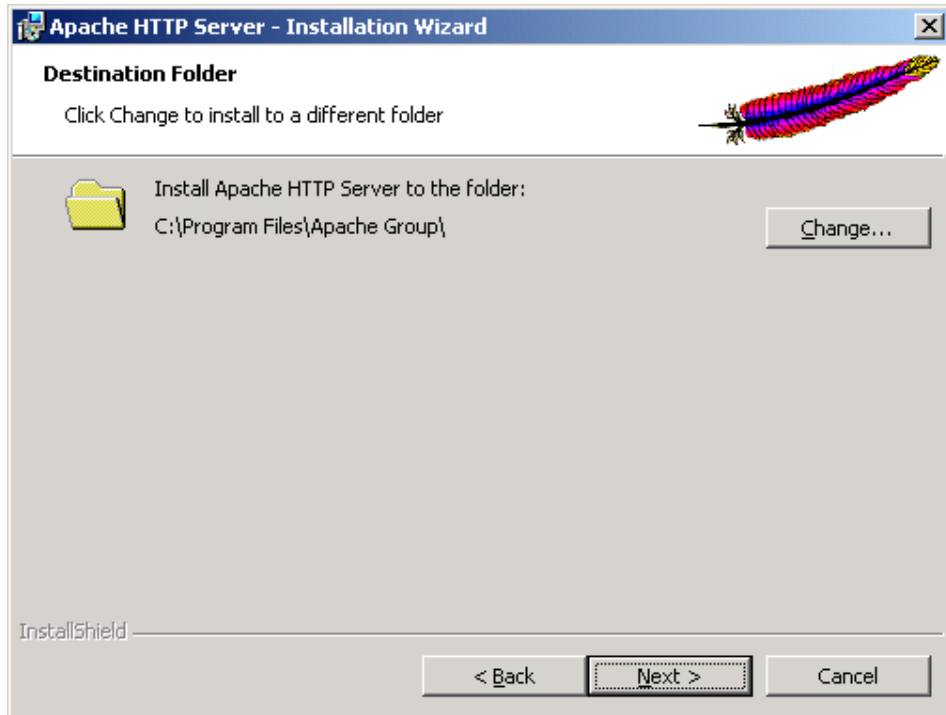
Exemple de configuration que j'ai fait pour ce tutorial...une fois la config' faite, cliquer sur le bouton « *Next* »...

Ici, nous avons le choix entre deux types d'installation, complète (**Complete**) ou personnalisée (**Custom**) :



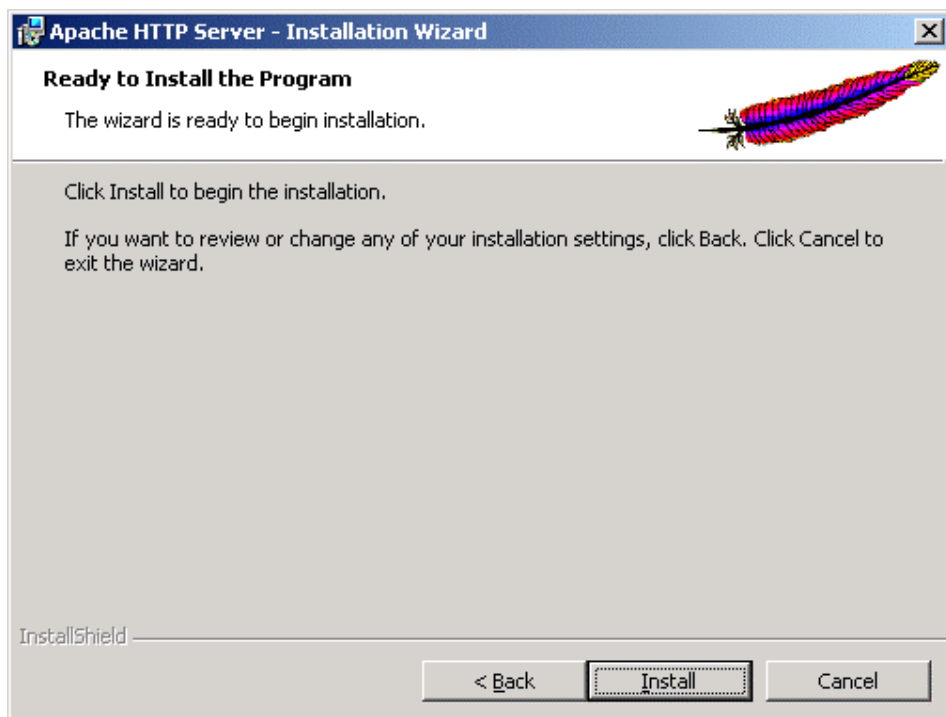
Nous faisons le choix de l'installation **Complete**, l'installation de type **Custom** permet de définir quelques petites choses : le répertoire dans lequel sera installé apache...ainsi que les options d'installation courante.

Cliquer sur le bouton « *Next* »



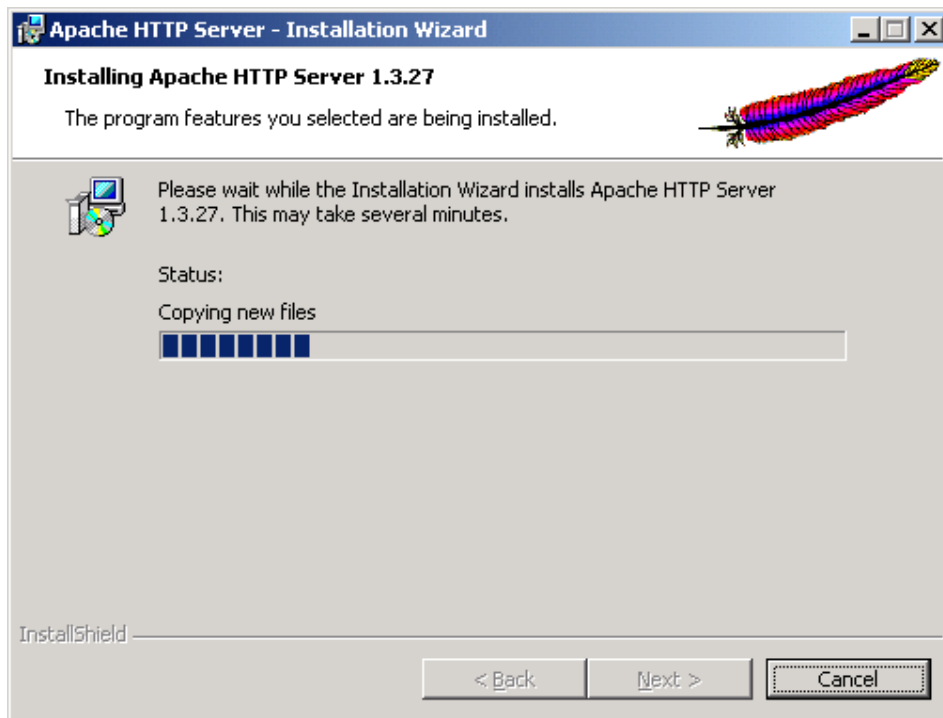
On choisi le répertoire d'installation puis on clique sur le bouton « *Next* »... (dans le cas présent, je le laisse **par défaut**...C:\Program Files\Apache Group\)

L'écran qui nous dit que l'installation peut commencer...



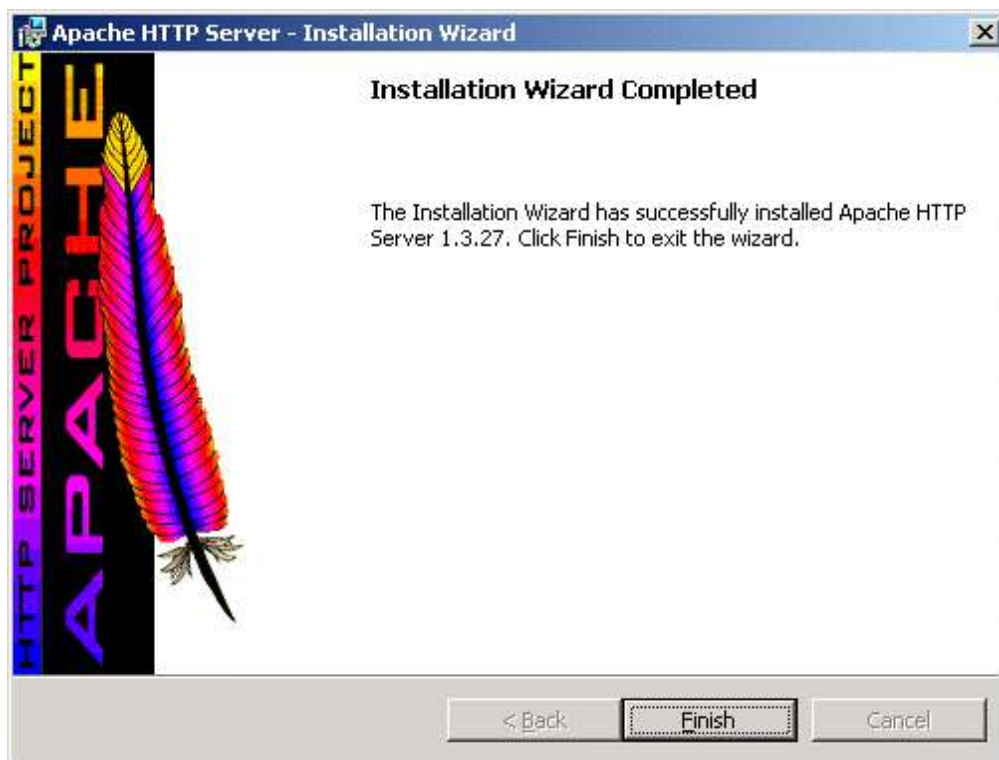
Pour débiter l'installation, je clique sur le bouton « *Install* »...

Installation en cours...



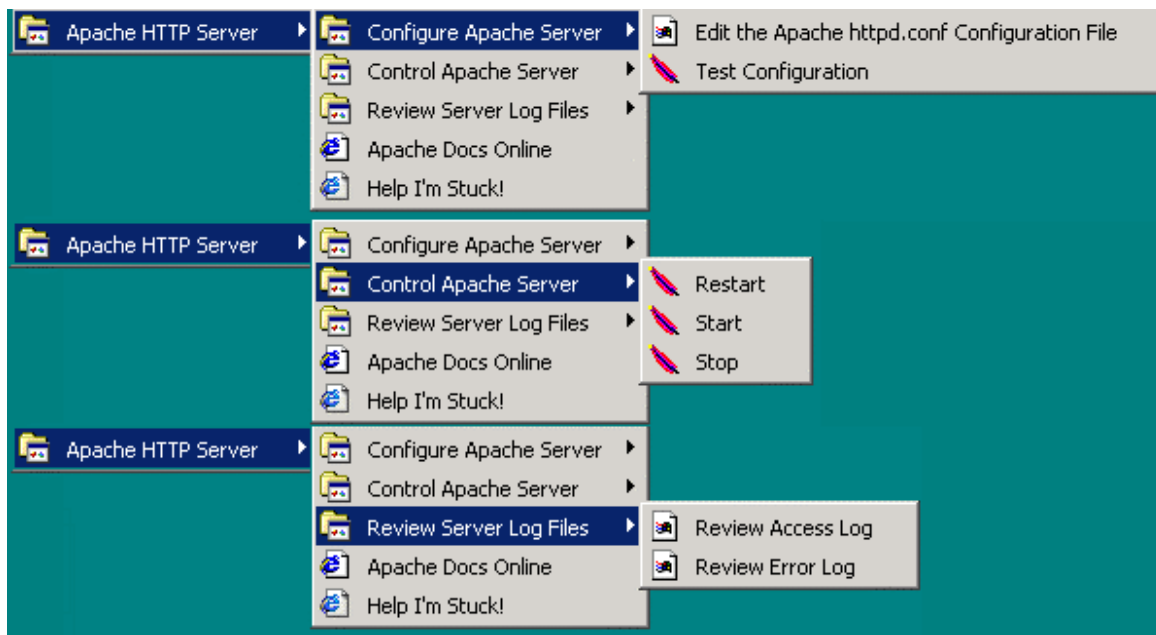
Si vous souhaitez annuler cliquer sur le bouton « **Cancel** »

Et voilà notre installation est à présent terminée...



Cliquer sur le bouton « **Finish** »...

Normalement si tout c'est bien passé (à l'installation) vous devriez avoir un menu similaire à celui ci-dessous (dans menu Démarrer/Programmes) :



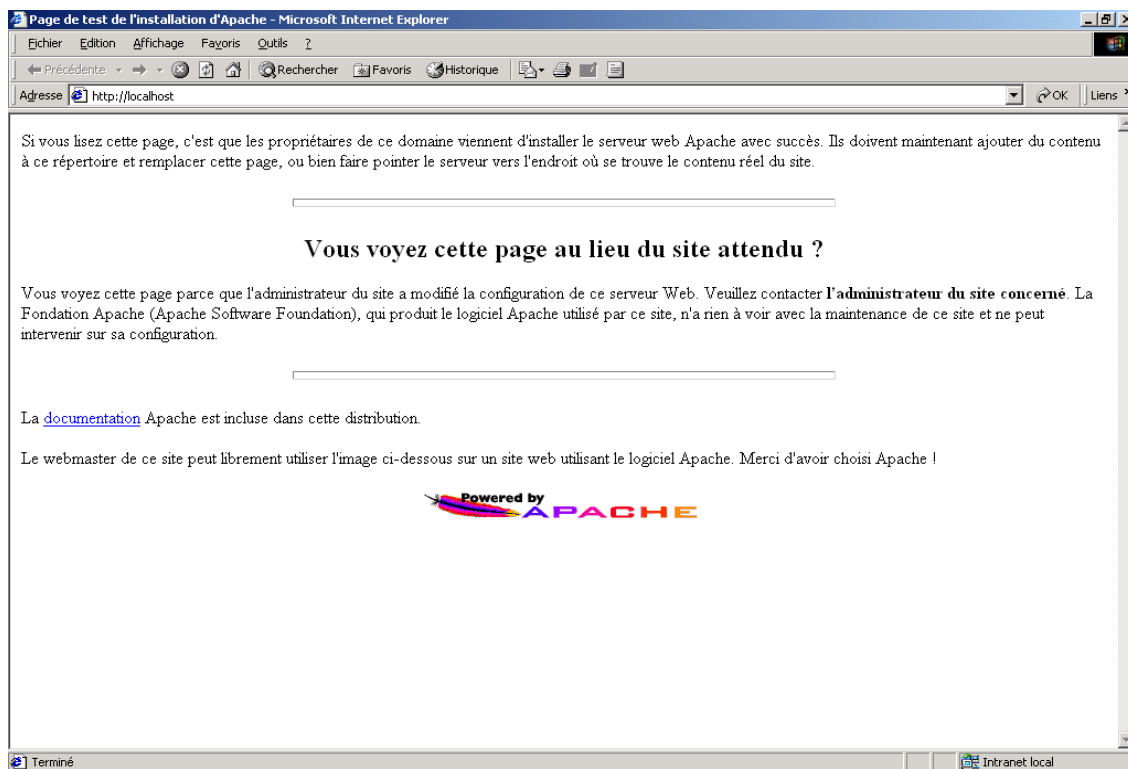
Ce menu nous permet d'avoir accès aux principaux réglages de Apache...

NOTE : Normalement une fois l'installation terminée, le serveur a été démarrée automatiquement...

1^{er} test de notre nouveau serveur Web !

Tapez dans votre navigateur (préférer) <http://localhost> (localhost correspondant au nom local de votre ordinateur)

Là, si tout c'est bien passer, vous devriez avoir un écran similaire à celui ci-dessous :



Si cette page s'affiche c'est que vous avez réussi...**BRAVO !!!**

Le 1^{er} test passé, nous allons maintenant voir les paramètres de configuration de notre serveur...

Pour ce faire nous allons stopper notre serveur, pour ce faire :

Cliquer sur **Démarrer/Programmes/Apache HTTP Server/Control Apache Server/Stop**

Là, votre serveur est à présent arrêté...

Pour configurer le serveur Apache, il vous suffit de cliquer sur

**Démarrer/Programmes/Apache HTTP Server/Configure Apache Server/
Edit the Apache httpd.conf Configuration File**

Une fois cliquer... le bloc-notes s'ouvre pour vous laissez entrevoir le fichier de configuration d'Apache. Un petit conseil, faite une copie de sauvegarde de votre fichier **httpd.conf** (qui se trouve **C:\Program Files\Apache Group\Apache\conf**, si vous avez effectuer l'installation par défaut).

Le fichier de configuration httpd.conf :

Le fichier **httpd.conf** est un fichier qui contient les différentes options de votre serveur apache, donc attention, faire attention à ce que vous modifiez si vous ne savez pas à quoi ça sert... A noter que dans notre fichier de configuration, certaine ligne commence par #, cela signifie que la ligne n'est pas pris en compte dans la configuration du serveur et qu'elle est en commentaire. Voyons maintenant quelques paramètres que vous pouvez trouver dans le fichier :

Ceci permet de définir le chemin où se trouve notre serveur apache...ici c'est le chemin par défaut.

ServerRoot "C:/Program Files/Apache Group/Apache"

Cette ligne bien que en commentaire par défaut, elle permet d'afficher le statut de mise en route de votre serveur ainsi que les requêtes qui lui ont été adressées.

#LoadModule status_module modules/mod_status.so (cette ligne fait appel au module mod_status.so)

Donc si vous souhaitez afficher les "stats" de votre serveur, enlever # (dé commenté) de cette ligne...

Cette ligne indique le port qu'utilise votre serveur Apache. Si vous souhaitez utiliser un autre port que celui par défaut, vous remplacez **80** par votre port, attention à ce que le port que vous utiliser ne soit pas déjà utilisé.

Port 80 (Port par défaut)

Cette ligne permet d'indiquer l'adresse email de l'administrateur du serveur (ce que l'on a défini tout à l'heure lors de l'installation).

ServerAdmin (dans le cas présent on trouve Maïsse.sebastien@cyberweb.net, ce que j'avais mis à l'installation)

Cette ligne permet d'indiquer le nom du serveur, si votre serveur est destiné a être juste utiliser en local, vous devez l'appeler **localhost**.

ServerName (définition du nom de votre serveur)

Cette ligne est "importante" car elle permet de définir le chemin de base de votre serveur. Ce qui permet de définir le répertoire dans lequel vos pages Web seront mis. Dans le cas présent (et par défaut), le répertoire est **"C:/Program Files/Apache Group/Apache/htdocs"**

DocumentRoot (répertoire principal de votre serveur)

C'est quelques lignes (par défaut) permettent de protéger l'accès votre serveur Web :

```
<Directory "C:/Program Files/Apache Group/Apache/htdocs">  
  Options Indexes FollowSymLinks MultiViews  
  AllowOverride None  
  Order allow,deny  
  Allow from all  
</Directory>
```

Dans le cas ci-dessus, étant la configuration par défaut d'accès au serveur, tout le monde peut avoir accès à serveur Web, si vous souhaitez interdire l'accès à tous les utilisateurs sauf à vous-même (127.0.0.1, l'IP loopback), il vous suffit de mettre un configuration similaire à celle ci-dessous :

Tout d'abord nous avons le répertoire auquel s'applique les restrictions d'accès (**C:/Program Files/Apache Group/Apache/htdocs** étant le répertoire par défaut dans le cas présent).

```
<Directory "C:/Program Files/Apache Group/Apache/htdocs">
```

```
#
```

```
Options Indexes FollowSymLinks MultiViews
```

```
# Cela permet de prendre en compte les différents style d'accès au répertoire quand il y a un .htaccess dans le  
# répertoire qui est rendu en accès limité.
```

```
AllowOverride None
```

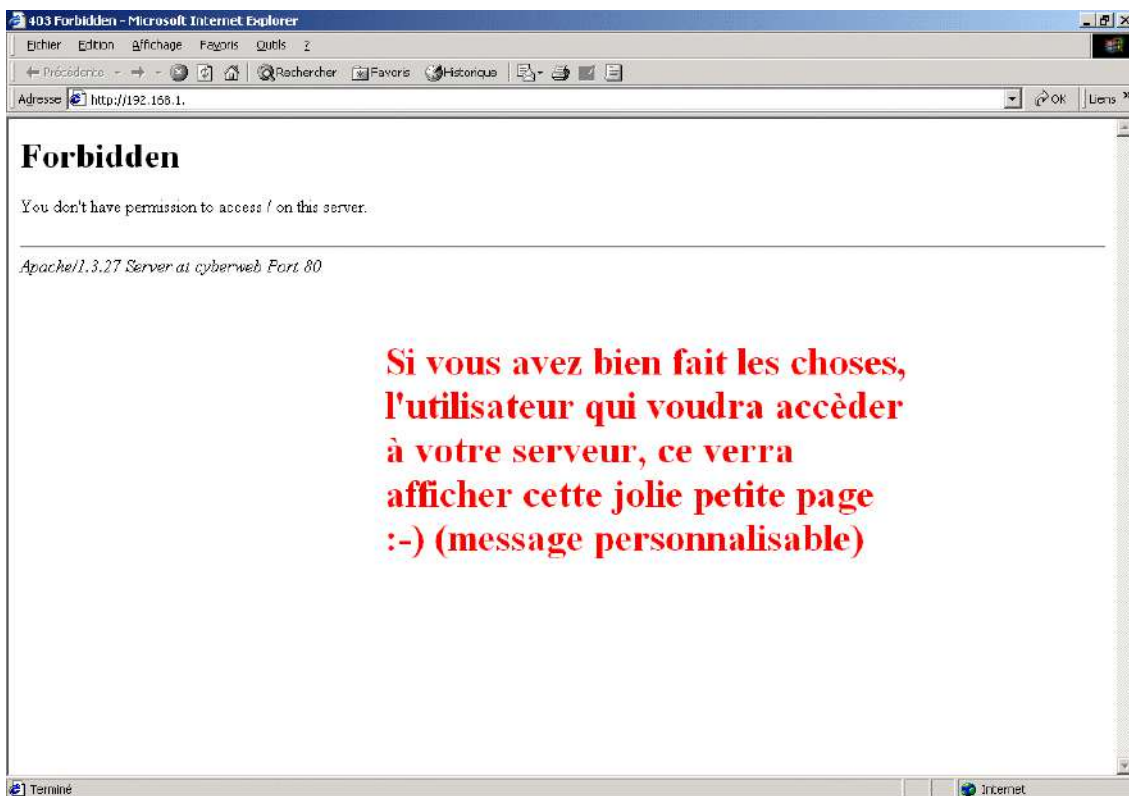
```
# L'ordre de prise en compte des « permissions ».
```

```
Order allow,deny
```

```
# Seul l'IP 127.0.0.1 est autorisée à accéder.
```

```
Allow from 127.0.0.1
```

```
</Directory>
```



Résultat de la protection du répertoire de base de votre serveur...**Erreur 403 !**

Cette ligne permet de définir la page qui sera pris par défaut par le serveur, dans la cas présent, c'est la page **index.html** qui est pris en considération.

DirectoryIndex index.html (si vous souhaitez en mettre plusieurs, vous devez mettre un espace)

Cette ligne permet de définir le nom du fichier qui sert à limiter en accès un répertoire (par défaut .htaccess). Ceci permettant ainsi de « verrouiller » un répertoire pour n'en autoriser que son utilisation aux personnes autorisées.

AccessFileName .htaccess

Cette partie permet de définir les autorisations sur un répertoire donné dans lequel on trouve un fichier .htaccess. Cela fonctionne sur le même principe que le faire de restreindre à une ou plusieurs IP l'accès au serveur... sauf que dans le cas présent nous pouvons avoir plusieurs authentifications pour accéder au répertoire.

```
<Files ~ "^\.ht">  
  Order allow,deny  
  Deny from all  
  Satisfy All  
</Files>
```

En ce qui concerne les messages qui sont affichés lorsqu'il y a une erreur sur le serveur (Error 403), peuvent se configurer de 3 manières différentes :

1^{er} Méthode : on tape le texte que l'on veut afficher pour une erreur directement dans le fichier de configuration.

Exemple : **ErrorDocument 403** "Vous n'avez pas accès à ce serveur (" doit être mis au début du texte).

2^{ème} Méthode : on fait une redirection local sur une page Web qui est plus personnalisable puisque l'on affiche une page que l'on aura créer auparavant.

Exemple : **ErrorDocument 403 /pasaccs.html**

3^{ème} et dernière méthode : On peut faire une redirection extérieure pour afficher le message d'erreur.

Exemple : **ErrorDocument 403** <http://www.machin.com/erreur/noaccess.html>

Désolé, vous n'avez pas accès à ce serveur !

Petite démo de personnalisation des messages d'erreur.

Ceci dit, vous avez maintenant les connaissances de base pour configurer votre serveur Web

NOTE : Pour afficher le statut concernant votre serveur, vous dé commentez les lignes suivantes :

LoadModule status_module modules/mod_status.so (On charge le module adéquat.)

AddModule mod_status.c (On ajoute la ligne qui permet d'ajouter le module adéquate.)

<Location /server-status> (Puis on rend accessible en local le répertoire adéquate)

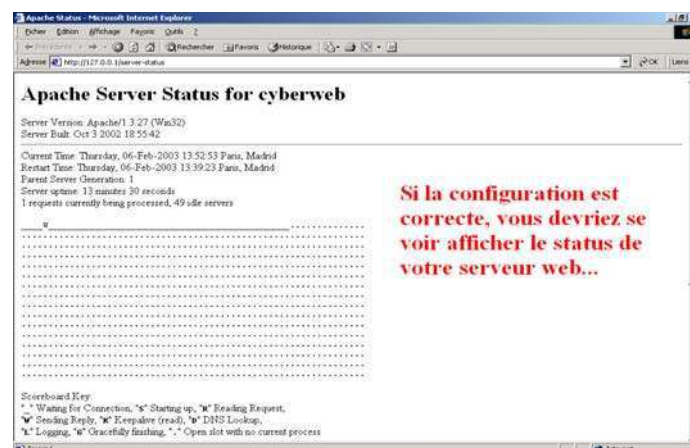
SetHandler server-status

Order deny,allow

Deny from all

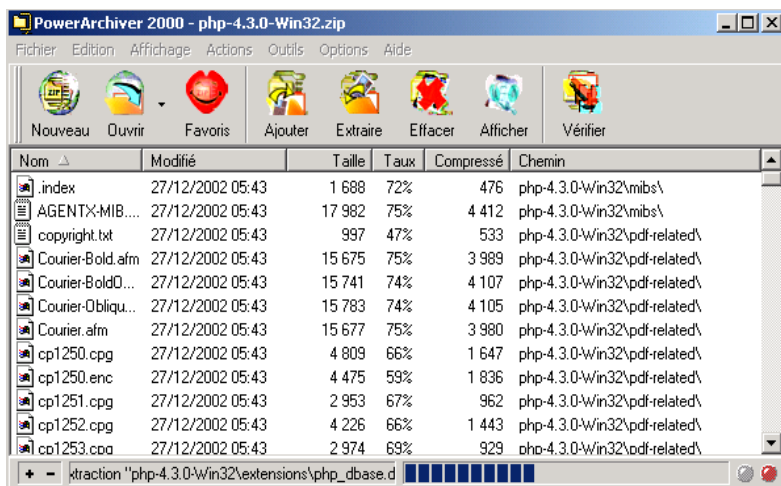
Allow from nom de l'ordinateur (127.0.0.1 dans le cas présent)

</Location>



INSTALLATION ET CONFIGURATION DE PHP

Avant de commencer il nous faut d'abord récupérer la dernière version de Php, cette dernière est la version 4.3.0, comme pour le serveur **Apache**, je vous conseil soit de vous rendre sur le site officiel <http://www.php.net> ou sur <http://www.geldenhuy.com>, de là, il vous faut vous rendre dans la rubrique **Dernières Versions** et télécharger **Php-4.3.0-Win32.zip** (taille ~ 6 Mo), une fois ceci fait, il vous faut décompresser le contenu de l'archive...



On décompresse le contenu de Php-4.3.0-Win32.zip

NOTE : je vous conseil de renommer le nom du répertoire dans lequel a été décompresser php...cela pouvant être source de problème pour la suite des opérations.

Renommer **php-4.3.0-Win32** en **php4**

Chemin lors de la décompression de l'archive : **C:\php-4.3.0-Win32**

Chemin d'accès à php une fois que je l'ai eu renommer : **C:\php4**

Première chose, il vous faut faire une copie de sauvegarde du **php.ini-inst** qui se trouve à la racine de votre répertoire de php...dans le cas présent **C:\php4**, une fois la copie de sauvegarde effectué, vous devez copier le fichier **php.ini-inst** puis le renommer en **php.ini** dans le répertoire Windows (donc **c:\windows** pour Windows 98 ou **c:\winnt** pour Windows 2000/XP/NT). Une fois ceci fait, vous éditez le fichier dans le bloc-notes, là, nous allons configurer quelques options. Lorsque le bloc-notes est lancé, faite une recherche pour trouver **extension_dir**, cette option permet de spécifier le répertoire dans lequel se trouve les dll adéquate pour le fonctionnement de php.

```
; Directory in which the loadable extensions (modules) reside.  
extension_dir = ./
```

Valeur par défaut de **extension_dir**

```
; Directory in which the loadable extensions (modules) reside.  
extension_dir = c:\php4
```

Comme vous pouvez le voir j'ai modifié la valeur de **extension_dir** pour mettre mon répertoire où se trouve php

Ceci fait, maintenant vous recherchez **extension=php**, normalement vous devriez en compter 45~, ces différentes lignes servent à charger les dll dont vous pourriez avoir besoin et qui serait alors charger au lancement de php, dans le cas présent on va laisser le tout en commentaire... vu que l'on a, pour le moment, pas besoin de ces dll.

NOTE : le caractère qui sert à mettre en commentaire est ;

```
extension=php_bz2.dll
extension=php_cpdf.dll
extension=php_crack.dll
extension=php_cur1.dll
extension=php_db.dll
extension=php_dba.dll
extension=php_dbase.dll
extension=php_dbx.dll
extension=php_domxml.dll
extension=php_exif.dll
extension=php_fbsql.dll
extension=php_fdf.dll
extension=php_filepro.dll
extension=php_gd.dll
extension=php_gd2.dll
extension=php_gettext.dll
extension=php_hyperwave.dll
extension=php_iconv.dll
extension=php_ifx.dll
extension=php_isfunc.dll
extension=php_imap.dll
extension=php_interbase.dll
extension=php_java.dll
extension=php_ldap.dll
extension=php_mbstring.dll
extension=php_mcrypt.dll
extension=php_mhash.dll
extension=php_mime_magic.dll
extension=php_ming.dll
extension=php_mssql.dll
extension=php_mysql.dll
extension=php_oc18.dll
extension=php_openssl.dll
extension=php_oracle.dll
extension=php_pdf.dll
extension=php_pgsql.dll
extension=php_printer.dll
extension=php_shmop.dll
extension=php_snmp.dll
extension=php_sockets.dll
extension=php_sybase_ct.dll
extension=php_w32api.dll
extension=php_xmlrpc.dll
extension=php_xslt.dll
extension=php_yaz.dll
extension=php_zip.dll
```

LISTE DES EXTENSIONS QUI SONT DISPONIBLES.

Pour plus d'informations sur ces différentes dll, je vous convie à vous rendre sur <http://www.php.net/manual/en/install-windows.php#install.windows.extensions>

Une fois notre fichier **php.ini-inst** enregistrer et puis fermer, nous pouvons retourner à notre fichier de configuration **Apache** (httpd.conf), là, nous allons configurer pour que nous puissions utiliser php. Donc, on édite le fichier de configuration, puis on ajoute les lignes suivantes au fichier :

```
LoadModule php4_module c:/php4/sapi/php4apache.dll
AddModule mod_php4.c
AddType application/x-httpd-php .php .php3 .php4
```

Vous prendrez aussi soin de modifier la valeur de **DirectoryIndex** pour que les pages dynamique soient prise en compte par votre serveur :

DirectoryIndex index.html (Avant modification)

Directory index.html index.php index.php3 index.php4 (Après modification)

Ceci va nous permettre de charger le module adéquat pour utiliser php avec notre serveur web.

```
LoadModule php4_module c:/le répertoire où vous avez décompactez/sapi/php4apache.dll
```

Si vous avez des problèmes de chargement de dll, message d'erreur qui peut s'afficher :

```
Cannot load c:/php/sapi/php4apache.dll into server
```

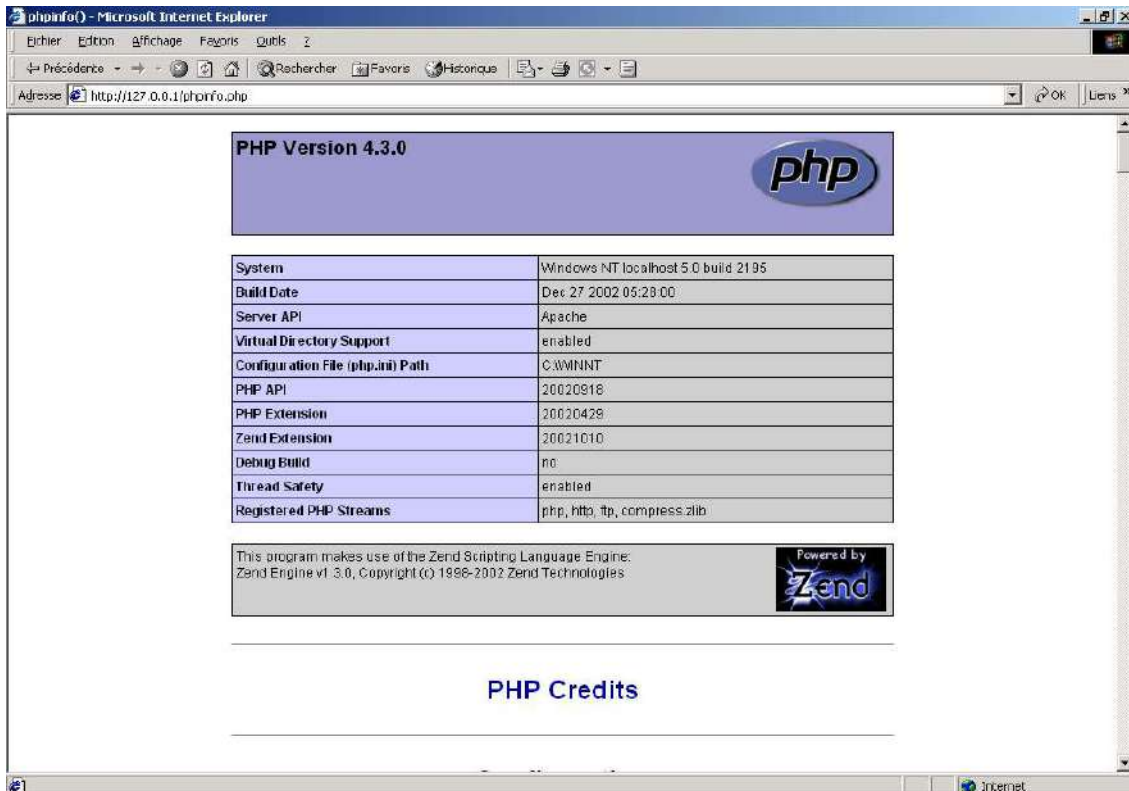
NOTE : Dans bien des cas le fait de faire une recherche puis de copier la dll dans le répertoire approprié permet de résoudre le problème. Copier le fichier **php4ts.dll** dans le répertoire **c:\winnt\system32** (dans le cas de Windows 2000, XP/NT) ou **c:\windows\system** (dans le cas de Windows 98).

Cela effectué, nous allons maintenant passer au test pour voir si notre configuration php fonctionne correctement. Pour cela, lancer un bloc-notes et taper les lignes suivantes puis sauvegarder sous **phpinfo.php** dans le répertoire de base de notre serveur Apache (dans le cas présent : **C:/Program Files/Apache Group/Apache/htdocs**) :

```
<?php
phpinfo();
?>
```

Ce code php va nous permettre d'afficher les options de configuration php du serveur.

Démarrer votre serveur Web et rendez-vous (dans votre navigateur) à <http://127.0.0.1/phpinfo.php>, si vous avez bien tout configuré, vous devriez voir apparaître un écran similaire à ce qui est affiché ci-dessous :



La page qui affiche les différents paramètres de votre php...**Bravo vous avez réussi !!!**

Voilà, voilà, à vous les joies de la programmation de vos pages Web en php...

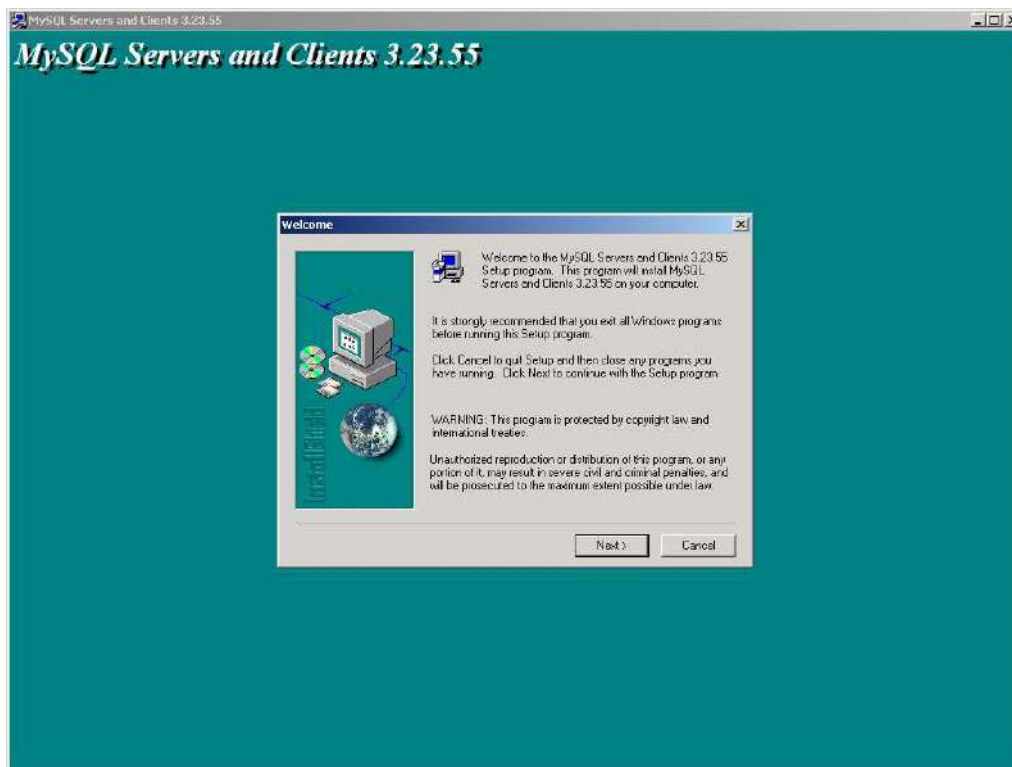
INSTALLATION ET CONFIGURATION DE MYSQL

Ce comme depuis le début, je vous joint à vous rendre sur le site officiel <http://www.mysql.com> ou sur <http://www.geldenhuy.com> (rubrique **Dernières Versions**) et à télécharger la dernière version de **mysql**, c'est-à-dire la version **mysql-3.23.55-win.zip** (taille ~ 13,7 Mo), une fois le téléchargement terminé...cliquez sur le fichier, là, le fichier va s'ouvrir et le cas échéant demander si vous souhaitez l'installer (dans le cas présent).



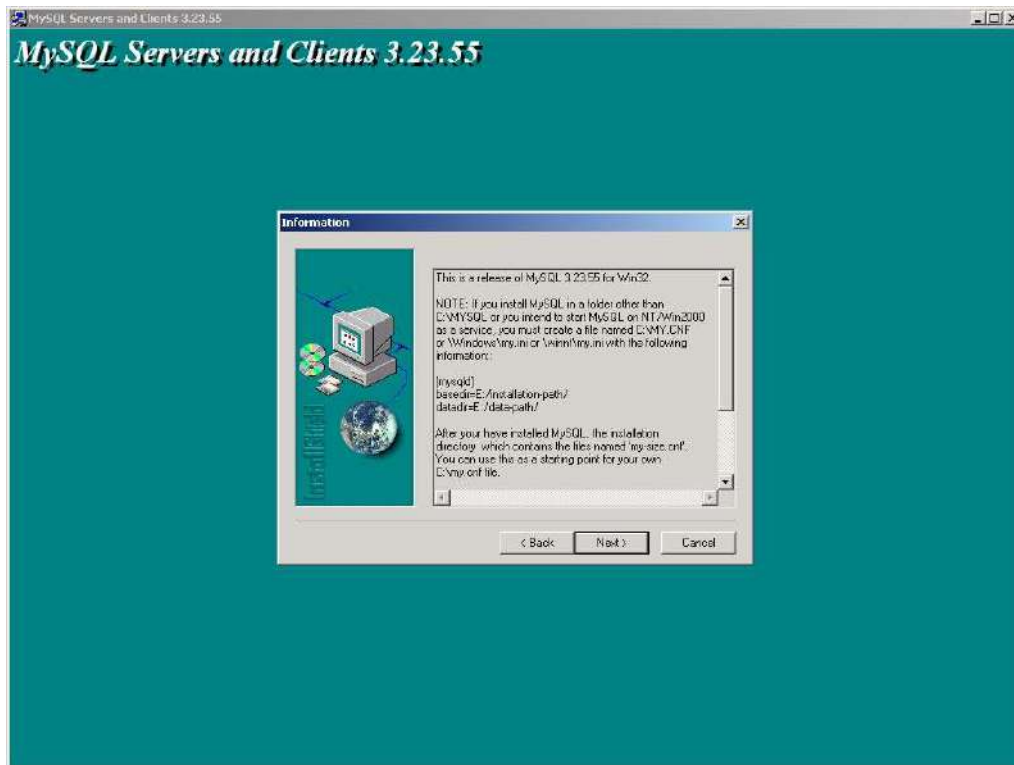
J'ai le choix entre soit installer soit de décompresser l'archive.

Je choisis d'installer directement **mysql**...



1^{er} écran de l'installation de mysql, cliquer sur le bouton « *Next* »

Cette écran vous rappelle que vous devez créer un fichier **my.ini** si le répertoire d'installation est différent de **c:\mysql**, une fois lu et pris note, vous pouvez cliquer sur le bouton « **Next** »...



L'écran suivant vous propose soit de garder le répertoire par défaut (**c:\mysql**) ou soit d'effectuer l'installation dans un autre répertoire (**Browse**).

NOTE : Si le répertoire d'installation est différent de **c:\mysql** alors vous devrez créer (au choix) un fichier **c:\my.cnf**, un fichier **\windows\my.ini** (dans la cas de Windows 98) ou un fichier **\winnt\my.ini** (dans le cas de Windows 2000, XP/NT).

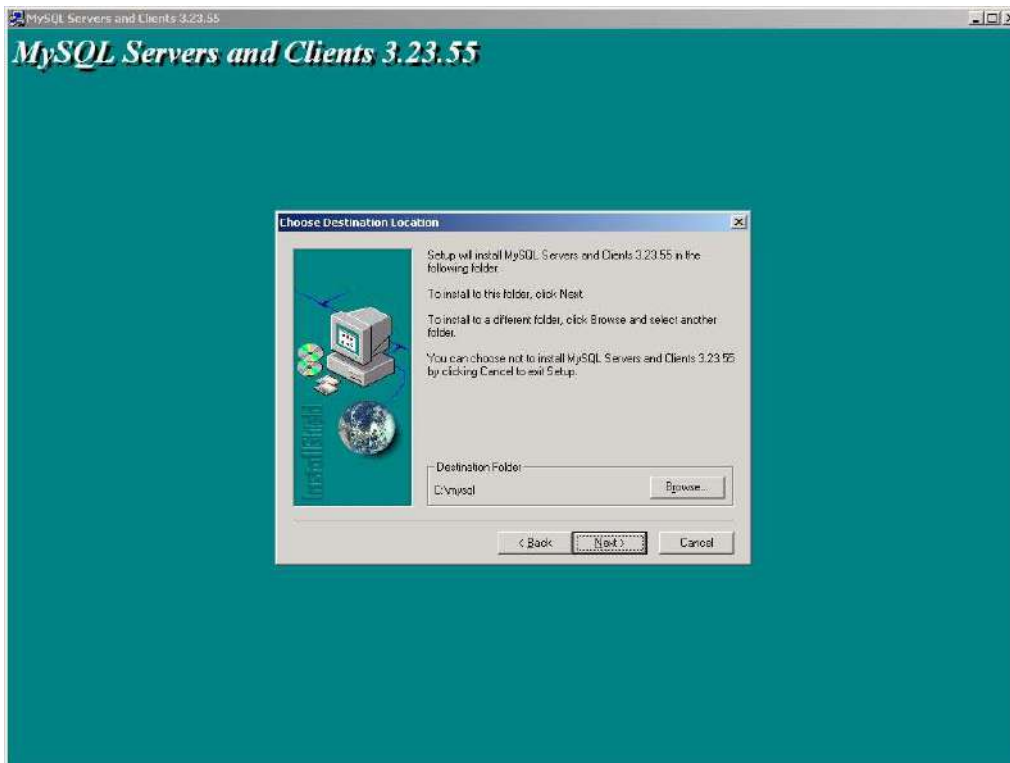
Ce fichier ayant comme contenu les informations suivantes :

[mysqld]

basedir=C:/installation-path/ (Le répertoire d'installation dans lequel vous avez installé mysql)

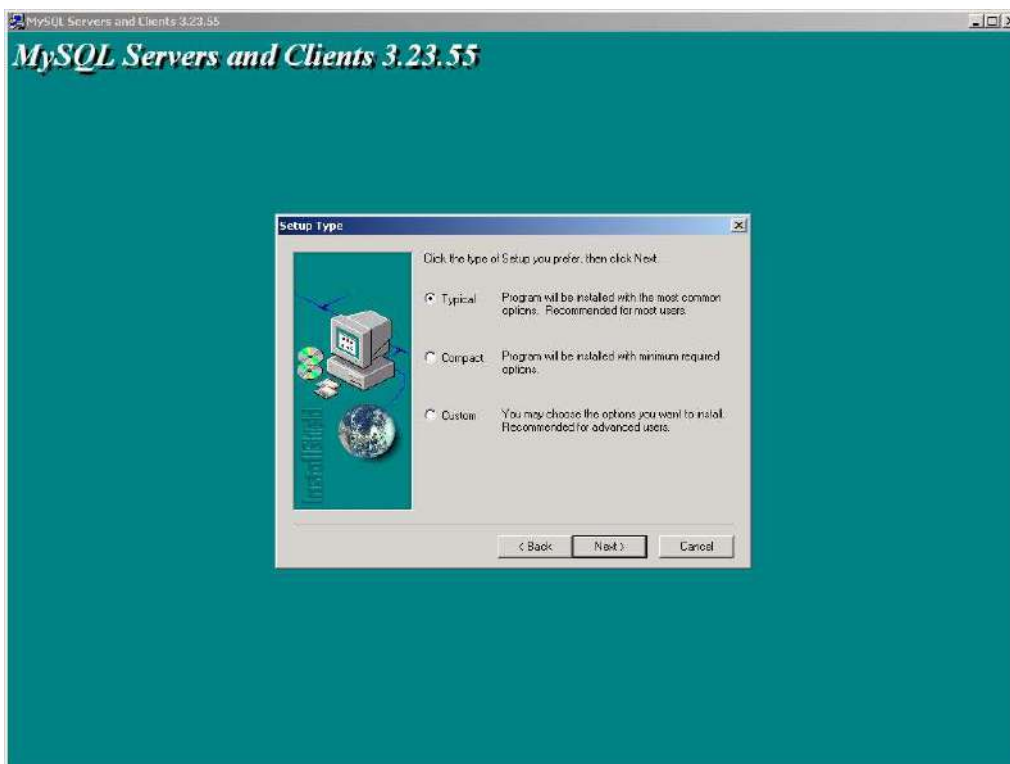
datadir=C:/data-path/ (Le répertoire contenant les données mysql)

Bien entendu vous pouvez installer mysql où bon vous semble du moment que ce dernier (le répertoire) est mentionné dans le fichier créé.



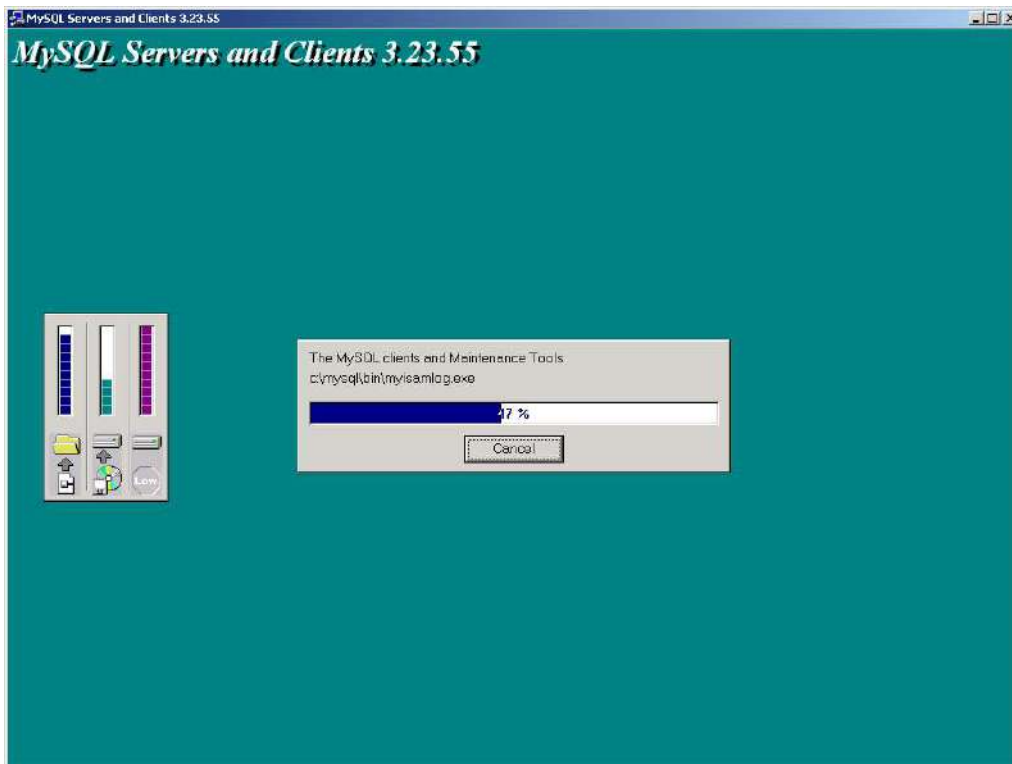
Dans le cas présent je vais effectuer l'installation avec le répertoire par défaut...
Donc ici on clique sur le bouton « *Next* » pour passer à l'étape suivante.

Ici, nous avons le choix entre différents styles d'installation...typique (**Typical**), compact (**Compact**) ou personnaliser (**Custom**). Typique étant l'installation qui est effectuée par la plupart des utilisateurs, la compact permet de n'installer que ce qui est nécessaire donc de prendre moins de place...et la personnaliser de laisser à l'utilisateur le choix de ce qui est installé.



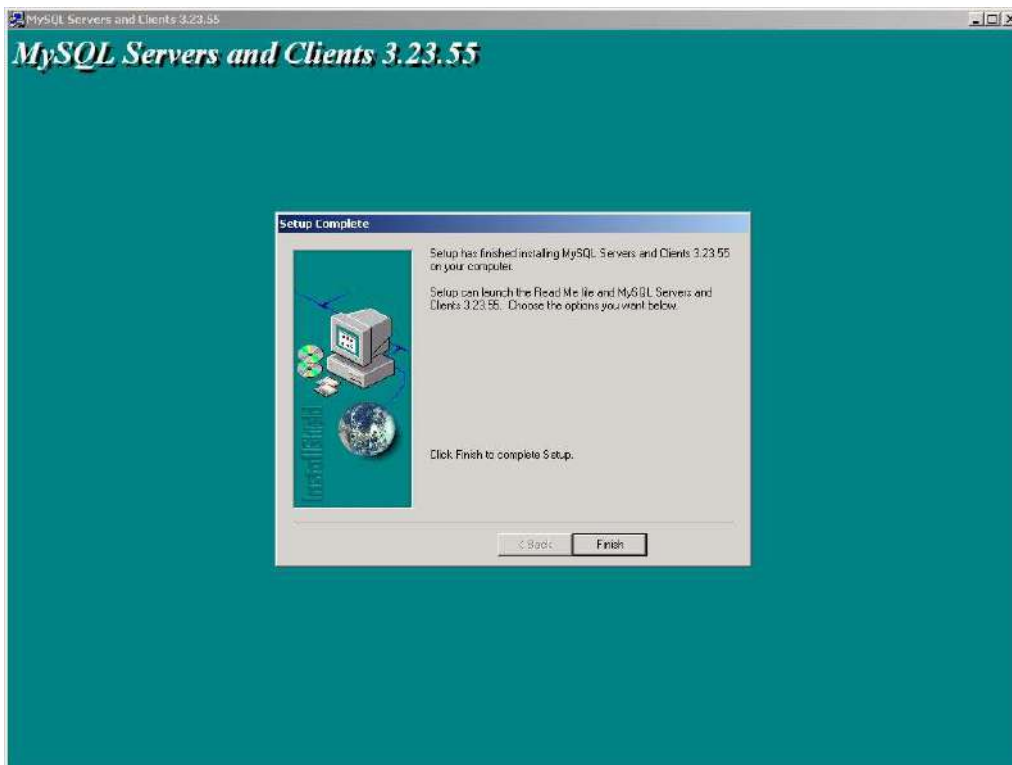
Dans le cas présent, nous allons faire une installation Typique (**Typical**)

Installation en cours...



Cliquer sur le bouton « *Cancel* » pour annuler l'installation.

L'installation est à présent terminée, cliquer sur le bouton « *Finish* » pour quitter cet écran :



Une fois l'installation terminée, nous allons voir comment on configure **mysql**, à noter que pour utiliser **mysql** nous avons soit la possibilité de le faire dans une invite de commande **Dos** (Démarrer/Exécuter/**cmd** ou **command** selon l'os que vous utilisez) soit par l'intermédiaire du programme **WinMySQLAdmin** qui se situe dans le répertoire (par défaut) **c:\mysql\bin**.

Si vous êtes sous Windows 2000/XP/NT, mysql est considéré comme un service (tout comme le serveur Apache) donc il démarre au lancement de l'ordinateur, ou si vous souhaitez le lancer par vous-même vous tapez :

Net start mysql (démarre le service mysql)

Net stop mysql (stop le service mysql)

```
C:\>net start mysql
Le service MySQL démarre.
Le service MySQL a démarré.

C:\>net stop mysql
Le service MySQL s'arrête.
Le service MySQL a été arrêté.
```

L'installation de ce service ne s'effectuant seul vous devrez taper l'un des deux commandes suivantes :

mysqld --install (le service démarre automatique)

mysqld --install --manual (le service démarre manuellement)

NOTE : En cas d'erreur 1067, veuillez à vérifier que vous avez bien créer le fichier **my.cnf** ou **my.ini** et qu'il se trouve à la bonne place (**c:** ou **\windows** ou **\winnt**).

Si vous êtes sous Windows 98, il vous suffit d'utiliser **WinMySQLAdmin** ou de taper la commande suivant :

C:\mysql\bin\mysqld (pour démarrer le serveur mysql)

C:\mysql\bin\mysqladmin -u root shutdown (pour stopper le serveur mysql)

Pour afficher le contenu de la base donnée (par défaut), on tape la commande suivante : **mysqlshow**

```
C:\mysql\bin>mysqlshow
+-----+
| Databases |
+-----+
| mysql    |
| test     |
+-----+
```

Contenu de la base (par défaut)

Pour définir un mot de passe pour l'administrateur (root) de la base mysql, il vous suffit de taper la commande suivante : **mysqladmin -u root password votremotdepasse**

Voilà pour les commandes de base de mysql, je vous laisse à présent consulter la documentation (en français) de mysql (http://dev.nexen.net/docs/mysql/annotee/manuel_toc.php).

INSTALLATION ET CONFIGURATION DE PHPMYADMIN

Voyons à présent la dernière étape de l'installation de notre serveur Web, ceci étant cette dernière étape est facultatif car elle n'est pas primordiale au bon fonctionnement de notre serveur, elle permet juste de faire en sorte que l'utilisation de mysql soit plus intuitif et ce à travers une interface Web...plutôt que d'utiliser des commandes pas toujours aisées à comprendre. Pour cette dernière étape il vous faudra donc vous rendre sur le site officiel de **phpmyadmin** <http://www.phpwizard.net> ou sur le site <http://www.geldenhuy.com> (rubrique **Dernières Versions**), la dernière version de phpmyadmin est la **phpMyAdmin-2.3.3pl1** (taille ~ 1,150 Mo), une fois cette dernière version récupérée, il vous faut décompresser son contenu.

Par défaut il se décompresse dans **c:\phpMyAdmin-2.3.3pl1**

Une fois décompresser, je le renomme en **c:\phpMyAdmin**

Editer le fichier **config.inc.php3**, dans ce dernier vous allez configurer le mot de passe que vous auriez pu mettre en compte **root**, ainsi que mettre la valeur de **\$cfg['PmaAbsoluteUri'] = 'http://127.0.0.1/mysql/';** (qui dans ce cas est <http://127.0.0.1/mysql/>). Une fois ceci fait, enregistrer le fichier avant de quitter... comme depuis le début de ce tutorial vous auriez pris l'habitude de faire une copie de sauvegarde du fichier **config.inc.php3**.

Ensuite nous éditons le fichier **httpd.conf** et aller dans la partie **<IfModule mod_alias.c>** de là, nous allons créer un alias qui nous permettra d'accéder à **phpmyadmin**, puis rajouter les lignes ci-dessous :

Alias /mysql/ "C:/phpMyAdmin/" (on crée l'alias **/mysql/** correspond donc à **c:/phpMyAdmin/**)

<Directory "C:/phpMyAdmin"> (On protège le répertoire pour qu'il ne soit accessible qu'à l'IP donnée)

Options Indexes MultiViews

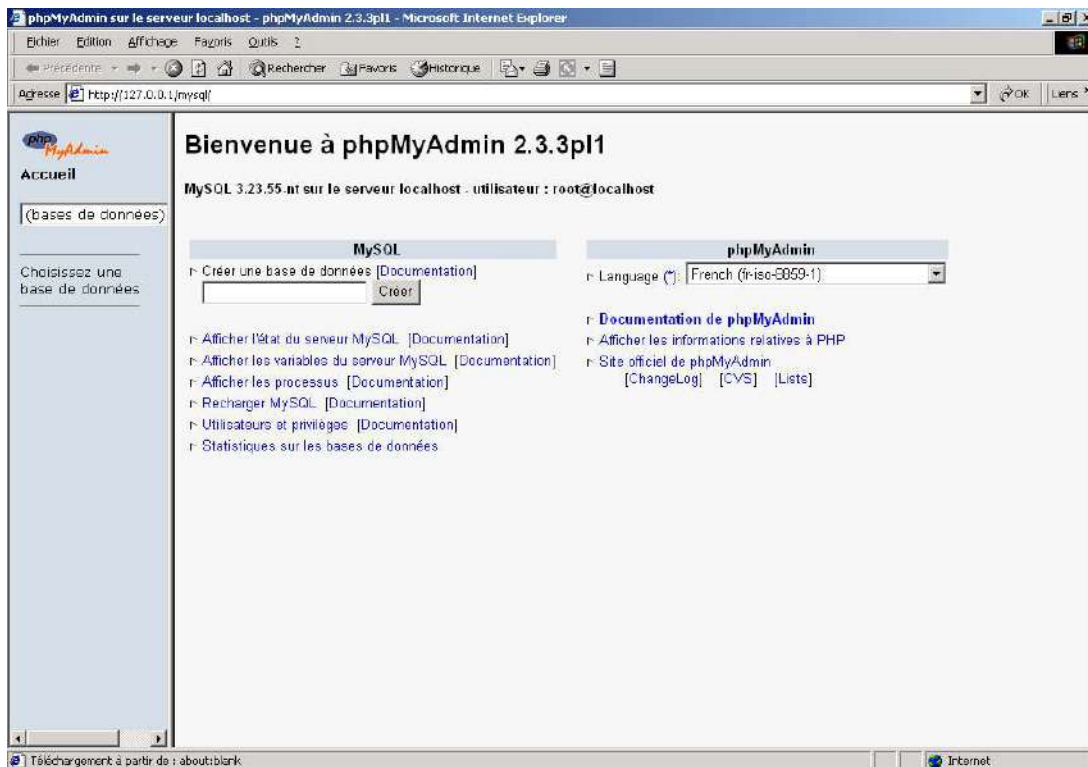
AllowOverride None

Order allow,deny

Allow from 127.0.0.1

</Directory>

Si tout se passe correctement, vous devez avoir un écran similaire à celui qui suit :



De là, vous pouvez créer, supprimer, modifier, etc... vos bases de données mysql.

INTRODUCTION A LA SECURITE D'UN SERVEUR

Lors de l'installation d'un serveur web, il vous faut avoir quelques notions de sécurité afin de protéger, un minimum, votre serveur d'éventuel intrus. Pour ce faire nous allons voir quelques options utiles.

NOTE : *Je ne prétend pas avoir les connaissances suffisantes pour mettre en place un serveur 100% sécurisé (car la sécurité est une chose relatif).*

Qu'est-ce que les fichiers .htaccess ?

Les fichiers .htaccess sont des fichiers de configuration Apache permettant de définir les règles dans un répertoire et sous répertoires (qui n'ont guère de tel fichier à l'intérieur). Pour cela on les utilise pour protéger un répertoire grâce à un mot de passe, ou pour interdire tout simplement l'accès à un répertoire donné.

Qu'elle est l'intérêt des fichiers .htaccess ?

L'intérêt des fichiers .htaccess est le fait qu'il peut-être utiliser dans n'importe quel répertoire virtuel ou sous répertoire du serveur.

Les principales raisons d'utilisation des fichiers **.htaccess** sont :

- Gérer l'accès à certains fichiers.
- Ajouter un mime-type.
- Protéger l'accès à un répertoire par un mot de passe.
- Protéger l'accès à un fichier par un mot de passe.
- Définir des pages d'erreurs personnalisées.

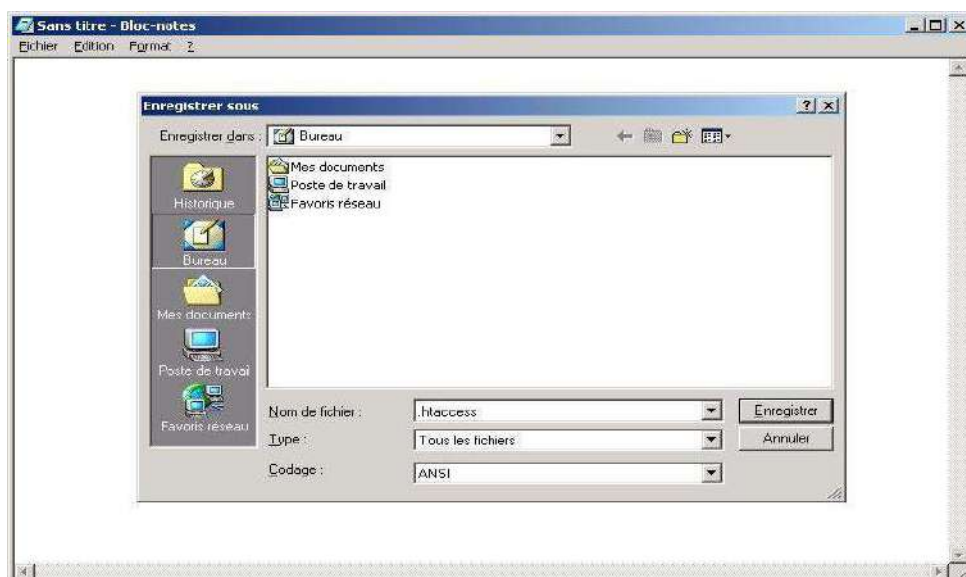
Principe de fonctionnement des fichiers .htaccess !

Le fichier **.htaccess** est placé dans le répertoire dans lequel il doit agir. Il agit ainsi sur les permissions du répertoire qui le contient et de tous ses sous-répertoires. Vous pouvez placer un autre fichier **.htaccess** dans un sous-répertoire d'un répertoire déjà contrôlé par un fichier **.htaccess**.

Le fichier **.htaccess** du répertoire parent reste en "activité" tant que les fonctionnalités n'ont pas été réécrites.

Sous Windows, il est logiquement impossible de créer un fichier **.htaccess**, puisque Windows ne vous autorisera pas à sauvegarder le fichier tel quel.

SOLUTION : Créer votre **.htaccess** avec le Bloc-notes (ci-dessous) ou tout autre éditeur supportant ce nom de fichier.



Protection d'un répertoire par mot de passe

La protection d'un répertoire par mot de passe est utile dans le cas où l'on souhaite mettre à disposition des documents ou faire une partie privative ne concernant que certaine personne. Dans ce cas, nous avons intérêt à protéger le dit répertoire par un login/pass qui sera fourni aux personnes habilités à venir.

La syntaxe d'un fichier **.htaccess** est la suivante :

```
AuthUserFile (emplacement du fichier de mot de passe)
AuthGroupFile (emplacement du fichier de groupe)
AuthName "Accès protégé"
AuthType Basic
<LIMIT GET POST>
Require valid-user
</LIMIT>
```

La commande **AuthUserFile** permet de définir l'emplacement du fichier contenant les logins et les mots de passe des utilisateurs autorisés à accéder à une ressource donnée.

La commande **AuthGroupFile** permet de définir l'emplacement du fichier contenant les groupes d'utilisateurs autorisés à s'identifier. Il est possible d'outrepasser cette déclaration en déclarant le fichier suivant : **/dev/null**.

Exemple de fichier **.htaccess** :

```
AuthUserFile /mon_site/mon_repertoire/.FichierPassword
AuthGroupFile /dev/null
AuthName "Accès protégé à mon site web"
AuthType Basic
<LIMIT GET POST>
Require valid-user
</LIMIT>
```

Crypter les mots de passe

Pour crypter les mots de passe d'un fichier **.htpasswd**. Apache est livré avec un utilitaire nommé **htpasswd** et qui est disponible dans le répertoire **bin** de votre répertoire d'installation d'Apache.

Le fichier de mot de passe (généralement nommé **.htpasswd**) est un fichier texte devant contenir sur chacune des ses lignes le nom de chaque utilisateur suivi des deux points (:) puis du mot de passe crypté (solution recommandée) ou en clair.

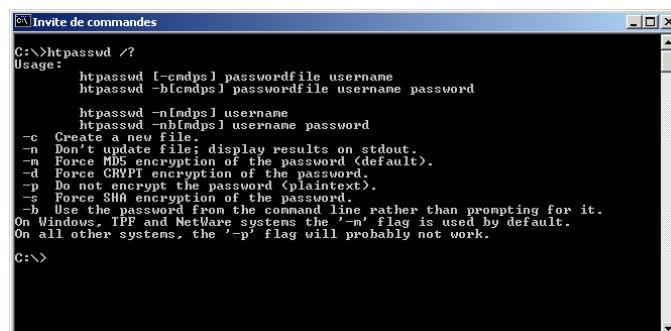
Ce fichier de mot de passe se doit être situer en dehors d'un répertoire virtuel internet ou si vous ne le pouvez pas (hébergement tiers), créer un répertoire étant lui même protéger et ayant pour seul accès le localhost.

Exemple de fichier **.htaccess** protégeant un répertoire contenant un **.htpasswd** :

Order deny, allow

Deny from all

Allow from 127.0.0.1 (localhost)



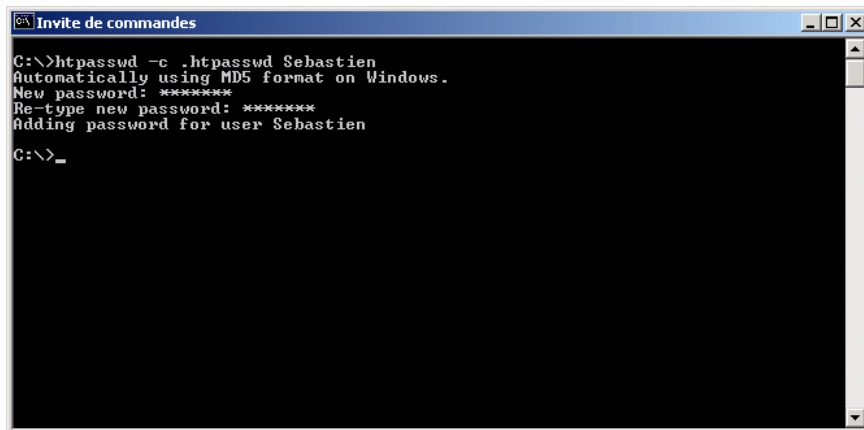
```
Invite de commandes
C:\>htpasswd /?
Usage:
    htpasswd [-cmdpsl] passwordfile username
    htpasswd -b[cmdpsl] passwordfile username password

    htpasswd -n[mdpsl] username
    htpasswd -nb[mdpsl] username password
-c Create a new file.
-n Don't update file; display results on stdout.
-m Force MD5 encryption of the password (default).
-d Force CRYPt encryption of the password.
-p Do not encrypt the password (plaintext).
-s Force SHN encryption of the password.
-b Use the password from the command line rather than prompting for it.
On Windows, ITP and NetWare systems the '-n' flag is used by default.
On all other systems, the '-p' flag will probably not work.
C:\>
```

La syntaxe (voir ci-dessus pour plus d'infos) de cet utilitaire est la suivante :

Pour créer un nouveau fichier de mots de passe :

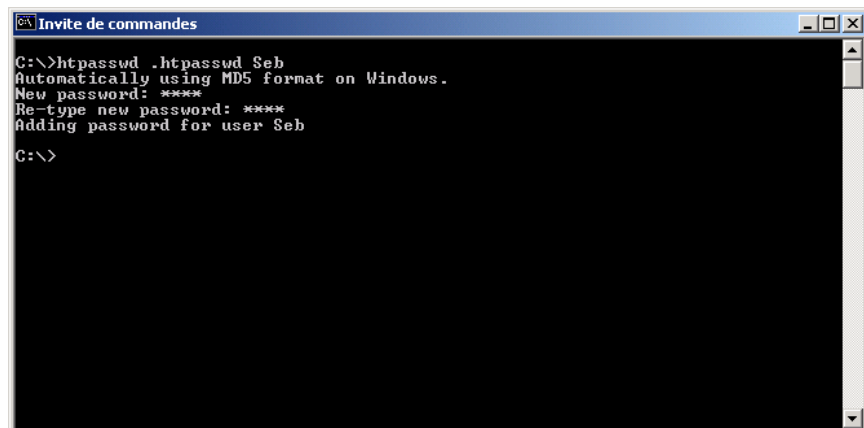
htpasswd -c (chemin du fichier de mot de passe) utilisateur



```
C:\>htpasswd -c .htpasswd Sebastien
Automatically using MD5 format on Windows.
New password: *****
Re-type new password: *****
Adding password for user Sebastien
C:\>_
```

Pour ajouter un nouvel utilisateur/mot de passe dans un fichier existant :

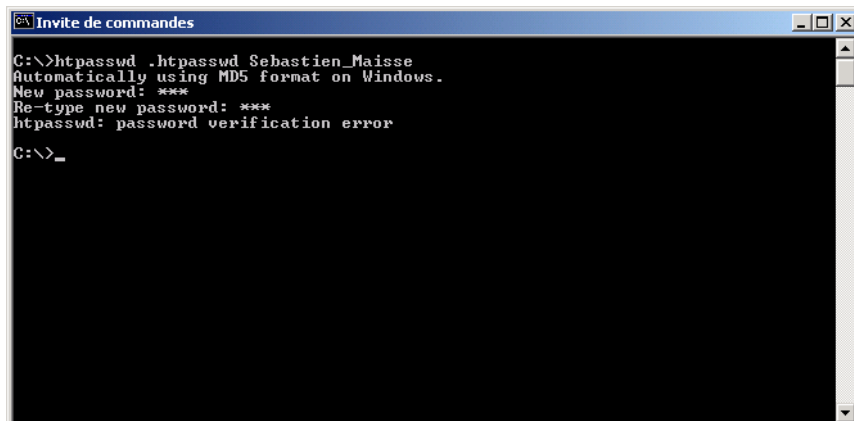
htpasswd (chemin du fichier de mot de passe) utilisateur



```
C:\>htpasswd .htpasswd Seb
Automatically using MD5 format on Windows.
New password: ****
Re-type new password: ****
Adding password for user Seb
C:\>
```

Le mot de passe sera demandé en ligne de commande avec une confirmation.

NOTE : Si vous ne confirmez pas correctement le mot de passe, vous aurez le message suivant :



```
C:\>htpasswd .htpasswd Sebastien_Maisse
Automatically using MD5 format on Windows.
New password: ***
Re-type new password: ***
htpasswd: password verification error
C:\>_
```

Là, l'utilisateur n'a pas pu être ajouté dans le fichier **.htpasswd**

Le résultat une fois les mots de passe cryptés pour les comptes créés précédemment :

```
Sebastien:$apr1$st3.....$R/EJlSkM8/JOz0zzX.77e0  
Seb:$apr1$.4.....$XLqQjJ93lFYPIWjFCeMxw/
```

Voilà donc une première approche concernant la sécurité d'un serveur web. Hormis la protection d'un répertoire par mot de passe il y a d'autres options de sécurité que l'on peut prendre en compte.

Autres options de sécurité

Comme par exemple la valeur de `ServerSignature` que l'on trouve dans le fichier **httpd.conf**. Cette valeur permet de définir ce qui est affiché par le serveur lors de problème (page non trouvé, etc...)

`ServerSignature` peut prendre 3 valeurs :

On : Affiche le nom du serveur (Apache X.X.XX)
Off : N'affiche pas le nom du serveur
Email : Affiche l'e-mail de l'administrateur du serveur.

`ServerTokens` peut prendre 3 valeurs également :

Prod(uctOnly) : Affiche juste le nom du serveur (Apache)
Min(imal) : Affiche le nom du serveur ainsi que sa version (Apache/X.X.XX)
OS : Affiche le nom du serveur, sa version ainsi que le système d'exploitation utilisé (Apache/X.X.XX (Unix))
Full : Affiche le maximum d'informations (Apache/X.X.XX (Unix) PHP/X.X MyMod/X.X)

Ces informations si elles sont fournies aux visiteurs permettent aux personnes malveillantes d'avoir déjà une bonne base pour préparer une éventuelle intrusion dans un système. Et ce sans qu'elle n'est eu à faire grand chose.

Il y a d'autres options que l'on peut prendre aussi en compte est qui concerne PHP. Voici donc certaines d'entre elles qui sont disponibles dans le fichier **php.ini**

Tout d'abord nous avons l'option **safe_mod** qui permet de vérifier que le propriétaire du script courant est le même que le propriétaire du fichier qui sera éventuellement manipulé par le script.

safe_mod = on (off étant la valeur par défaut)

Puis l'on a aussi l'option **display_errors** qui permet d'afficher à l'écran ou non les messages d'erreurs lorsqu'un script n'a pu s'exécuter correctement.

display_errors = off (on étant la valeur par défaut)

Et pour finir nous avons l'option **log_errors** qui permet d'enregistrer dans le fichier **error_log** de Apache les messages d'erreurs lorsqu'il y a une erreur d'exécution d'un script.

log_errors = on (off étant la valeur par défaut)

Comme vous avez pu le voir dans le présent document, il existe tout un panel de petites astuces qui permettent de rendre votre serveur web plus sécurisé que par défaut. Bien entendu cette liste d'options n'est pas exhaustive et je vous engage par la même à aller vous documenter sur le sujet sur les différents sites qui parlent de tout ça (voir ci-dessous pour les ressources).

Par ailleurs il vous est vivement conseillé de vous tenir au courant des dernières failles de sécurité concernant les applications que vous pourriez utiliser ainsi vous éviterez bien des soucis. Ce tenir informé est une des clés de la sécurité d'un serveur.

RESSOURCES

Une sélection de quelques url qui pourront vous être utiles pour compléter ce tutorial.

APACHE

<http://www.apache.org> (Site officiel d'Apache)

<http://www.apachefrance.com> (La version française du site)

PHP

<http://www.php.net> (Site officiel de PHP)

<http://www.phpfrance.com/> (Site sur le PHP en français)

<http://www.afup.org/> (Site de l'Association Française des Utilitaires de PHP)

MYSQL

<http://www.mysql.com> (Site officiel de MySQL)

<http://dev.nexen.net/docs/mysql/chargement.html> (Documentation en français de mysql)

PHPMYADMIN

<http://www.phpwizard.net/projects/phpMyAdmin/> (Site officiel du projet phpMyAdmin)

<http://pixi.free.fr/phpmyadmin.html> (Site concernant la configuration de base de phpMyAdmin)

http://www.phpindex.com/articles/articles_lire.php3?element=151 (Page qui vous explique la configuration avancée de phpMyAdmin)

SECURITE

<http://www.commentcamarche.net/apache/apacht.php3> (Tutorial dont je me suis inspiré concernant les fichiers .htaccess et .htpasswd)

CONCLUSION

Voilà, normalement après avoir suivi du début à la fin ce petit tutorial vous devriez être en mesure de mettre en œuvre un serveur Web qui vous permettra soit de créer et d'héberger votre site personnelle ou soit tout simplement de pouvoir tester différentes choses et ce en local sur votre ordinateur (sans pour autant avoir besoin d'une connexion à internet).

En espérant que ce tutorial aura pu vous aider (même un peu) dans la création de votre serveur Web, j'en suis heureux, si vous souhaitez me contacter pour me donner vos impressions sur ce tutorial ou pour avoir de plus amples informations sur l'un des sujets abordés, vous pouvez le faire à l'adresse e-mail suivante :

thecyberseb@hotmail.com

Sur ce à bientôt dans un prochain tutorial... :-)