

INSTALLATION ET CONFIGURATION D'UN SERVEUR WEB

(3ème édition)

Par **Maisse Sébastien**



Document en date du 22 novembre 2006

Sommaire

Concernant ce document :	3
Préambule :	4
Qu'est-ce qu'un serveur web ?	4
Téléchargement d'APACHE :	5
Installation d'APACHE :	5
Configuration d'APACHE :	18
Un mot sur les hôtes virtuels (Virtual Hosts) :	21
Qu'est-ce que PHP ?	23
Téléchargement de PHP :	23
Installation de PHP :	24
Un mot sur la sécurité :	28
Qu'est-ce qu'un serveur mysql ?	30
Téléchargement de MySQL :	31
Installation de MySQL :	32
Configuration de MySQL :	37
Quelques interfaces graphiques :	49
Rendre accessible son serveur web :	55
Ressources :	56
Quelques url's :	56
Quelques livres :	56
Conclusion :	57
Merci !	57

Concernant ce document :

Ce document a pour vocation de vous aider à mettre en place un serveur web sur une plateforme Microsoft Windows (version XP, 2000, 2003). Cette documentation est une mise à jour d'un tutoriel que j'avais rédigé il y a 3 ans de cela (en 2003 pour la première version du présent document).

Les précédentes versions de ce tutoriel sont toujours en ligne et rencontre un vif succès à l'heure d'aujourd'hui, d'où ma décision de réaliser une troisième édition prenant en compte les dernières versions des logiciels qui vont vous permettre de rendre disponible votre futur serveur internet.

Les logiciels que nous allons mettre en application sont tous disponibles gratuitement ou en open source. Ils sont les concurrents des logiciels commerciaux qui sont vendus par de grandes et puissantes sociétés informatiques avides de monopole pour les solutions qu'elles proposent.

Vous allez être accompagnés tout au long de ce tutoriel dans l'installation mais aussi la configuration de chaque logiciel ce qui vous permettra au final d'avoir un serveur web (Apache) supportant le langage PHP et pourvu d'un serveur de données (MySQL).

Si lors de l'exploitation de ce document vous rencontrez des erreurs, ou des incompréhensions vous pourrez obtenir de l'aide en fin de ce document.

Je vous souhaite à présent une bonne lecture ainsi qu'une bonne installation/configuration de votre futur serveur.

Préambule :

Bienvenue dans ce document qui a pour but de vous faire découvrir l'installation du serveur web Apache avec le support PHP ainsi que du serveur de données MySQL dans un environnement Windows (XP, 2000 ou 2003).

Apache est le travail de la **Apache Software Foundation** qui réunit d'autres projets.

Mais avant tout, une petite explication (provenant du site wikipedia.org) de ce qu'est un serveur web et à quoi il est destiné.

Qu'est-ce qu'un serveur web ?

Un serveur HTTP ou démon HTTP ou HTTPd (HTTP daemon) ou (moins précisément) serveur Web, est un logiciel servant des requêtes respectant le protocole de communication client-serveur HyperText Transfer Protocol (HTTP), qui a été développé pour le World Wide Web.

Un ordinateur sur lequel fonctionne un serveur HTTP est appelé serveur Web. Le terme « serveur Web » peut aussi désigner le serveur HTTP (le logiciel) lui-même. Les deux termes sont utilisés pour le logiciel car le protocole HTTP a été développé pour le Web et les pages Web sont en pratique toujours servies avec ce protocole. D'autres ressources du Web comme les fichiers à télécharger ou les flux audio ou vidéo sont en revanche fréquemment servies avec d'autres protocoles.

Les serveurs HTTP les plus utilisés sont :

- Apache HTTP Server de la Apache Software Foundation, successeur du NCSA httpd
- Internet Information Services de Microsoft (IIS)
- Sun ONE de Sun Microsystems (anciennement iPlanet de Netscape Communications Corporation)
- Le serveur Web Zeus de Zeus Technology

Le plus populaire est Apache HTTP Server qui sert environ 69% des sites Web en octobre 2005 selon Netcraft.

Historiquement, d'autres serveurs HTTP importants furent CERN httpd, développé par les inventeurs du Web, abandonné le 15 juillet 1996 et NCSA HTTPd, développé au NCSA en même temps que NCSA Mosaic, abandonné mi-1994.

Il existe aussi des serveurs HTTP qui sont des serveurs d'applications capables de faire serveur HTTP, comme Roxen ou Caudium.

Téléchargement d'APACHE :

Avant l'installation, nous devons tout d'abord récupérer l'archive contenant notre futur serveur web. Pour se faire rendez-vous sur le site [Apache.org](http://httpd.apache.org). De là, il faut récupérer la version 2.0.XX de Apache pour environnement Windows. Une troisième branche de développement a fait son apparition, la version **2.2.x** est désormais disponible, à l'heure de la rédaction de ce document aucun module php officiel n'est disponible pour cette version.

Dans le cas présent, nous utiliserons donc la version **2.0.59** d'Apache.

Nom du fichier téléchargé : **Apache_2.0.59-win32-x86-no_ssl.msi**

Bien entendu, il va de soi que vous pouvez baser votre installation sur une version supérieure. Vous pouvez effectuer le téléchargement de l'application à cette url :

<http://httpd.apache.org/download.cgi>

Ceci fait, nous pouvons à présent installer Apache sur notre machine.

Installation d'APACHE :

Localisez l'archive que vous avez téléchargée, par défaut sur le bureau si vous avez téléchargé avec Firefox sinon une petite recherche (sur votre disque dur) vous permettra de localiser le programme d'installation d'Apache.

Si vous avez déjà une version d'Apache d'installée, il y a de forte chance que l'écran ci-dessous s'affiche :

Dans ce cas, procéder à la désinstallation de la version installée avant de poursuivre l'installation de la nouvelle version. A noter que ce message ne s'affichera que si c'est une version de la branche 2.0.xx d'Apache.



Illustration 1: Le message d'erreur qui s'affiche si une version d'Apache est déjà installée.

NOTE : Si vous avez une version **2.0.55** d'Apache, il affichera le message, par-contre si vous avez une version **2.2.x**, il n'y aura pas de message d'erreur.

Un double-clic lancera le programme d'installation... On clique sur le bouton **Next**.



Illustration 2: Ecran d'introduction de l'installation d'Apache

Nous devons tout d'abord accepter la licence d'utilisation d'Apache. La licence est similaire à une licence [GPL](#).

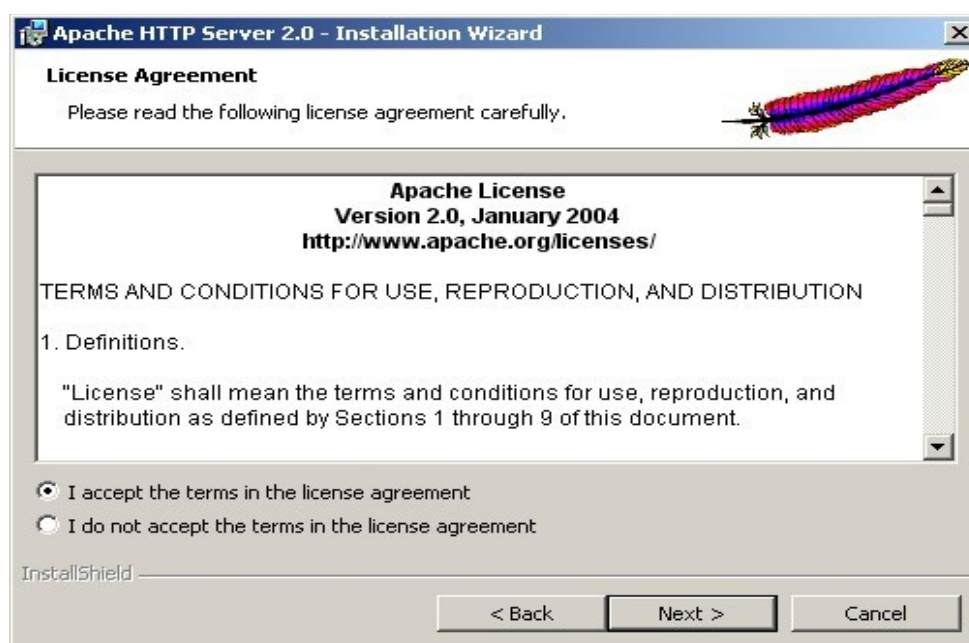


Illustration 3: Ecran d'acceptation de la licence d'Apache

On coche **I accept the terms in the licence agreement** puis on clique sur **Next**.

Ceci étant si vous n'acceptez pas les termes de la licence, l'installation s'achèvera là...

Une petite explication de ce qu'est Apache. On clique sur **Next**.

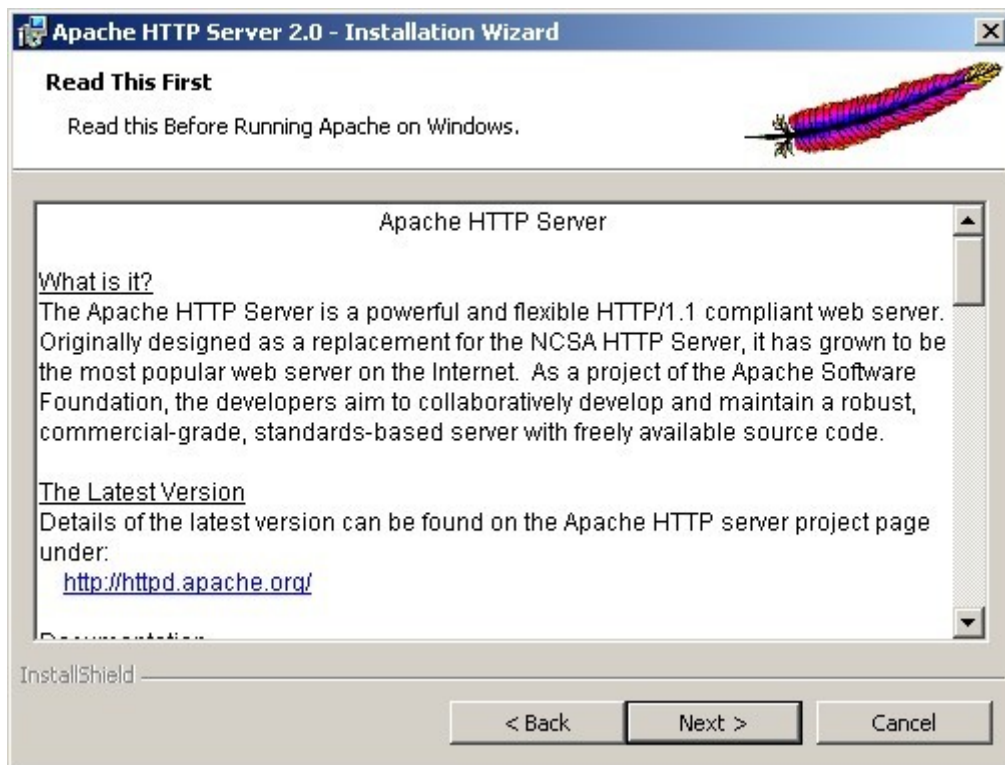


Illustration 4: Ecran qui fournit une explication de ce qu'est Apache

A cet écran, nous allons configurer le nom de domaine, le nom du serveur ainsi que l'adresse e-mail de l'administrateur du serveur.

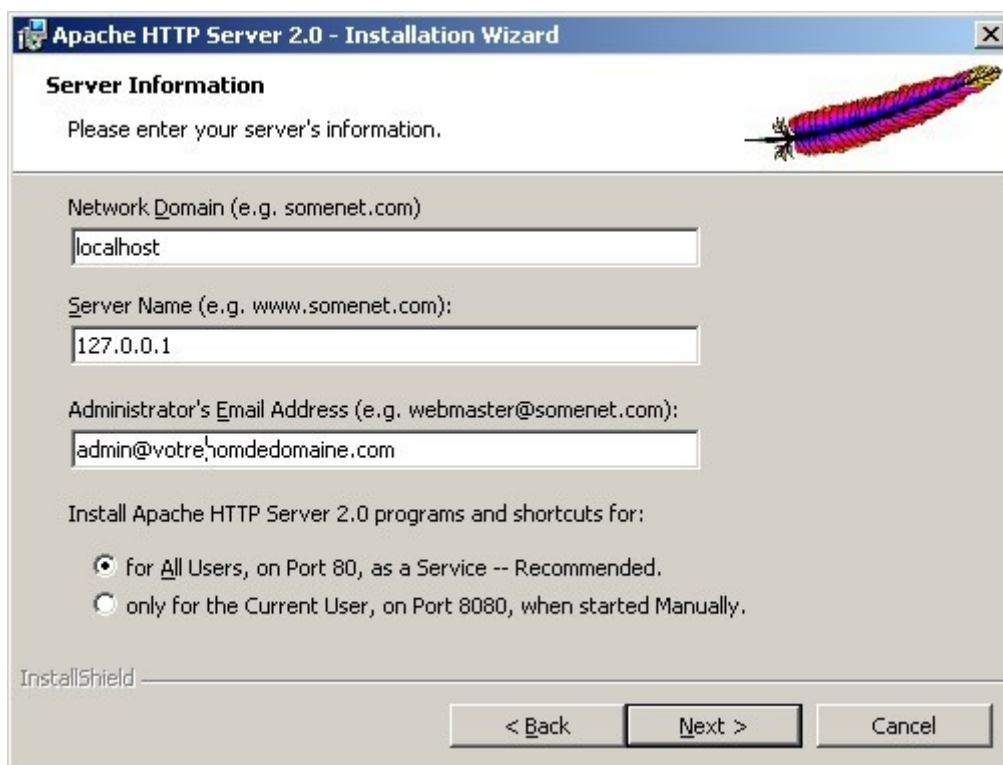


Illustration 5: Ecran de configuration du Network Domain / Server Name / e-mail Administrator

Network Domain : Nom de domaine du serveur, [Localhost](#) correspondant à l'hôte local dans le cas présent.

Server Name : Nom du serveur, ici, nous allons mettre 127.0.0.1 qui correspond à l'ip interne de la machine.

Administrator's Email Address : l'adresse e-mail de l'administrateur en l'occurrence vous.

For All Users, on Port 80, as Service : cette option permet d'installer Apache **pour tous les comptes utilisateurs** en utilisant le **port 80** (port utilisé par défaut pour un serveur web) **en tant que service** (le service est utilisé par Windows XP ou 2000). Pour ce qui concerne Windows 98, il faudra lancer manuellement le serveur puisque cette version de Windows ne dispose pas de service.

only for the Current User, on Port 8080, when started Manually : cette option permet d'installer Apache pour **l'utilisateur courant** (qui procède à l'installation) en utilisant le **port 8080** et avec un **démarrage manuel**. Cette dernière est recommandée si vous ne souhaitez pas mettre en production votre serveur web, mais que vous l'utilisez dans un objectif de développement web.

Dans le cas présent, je procède à une installation de type « tous utilisateurs, sur le port 80 en tant que service ».

On clique sur **Next** pour continuer l'installation...

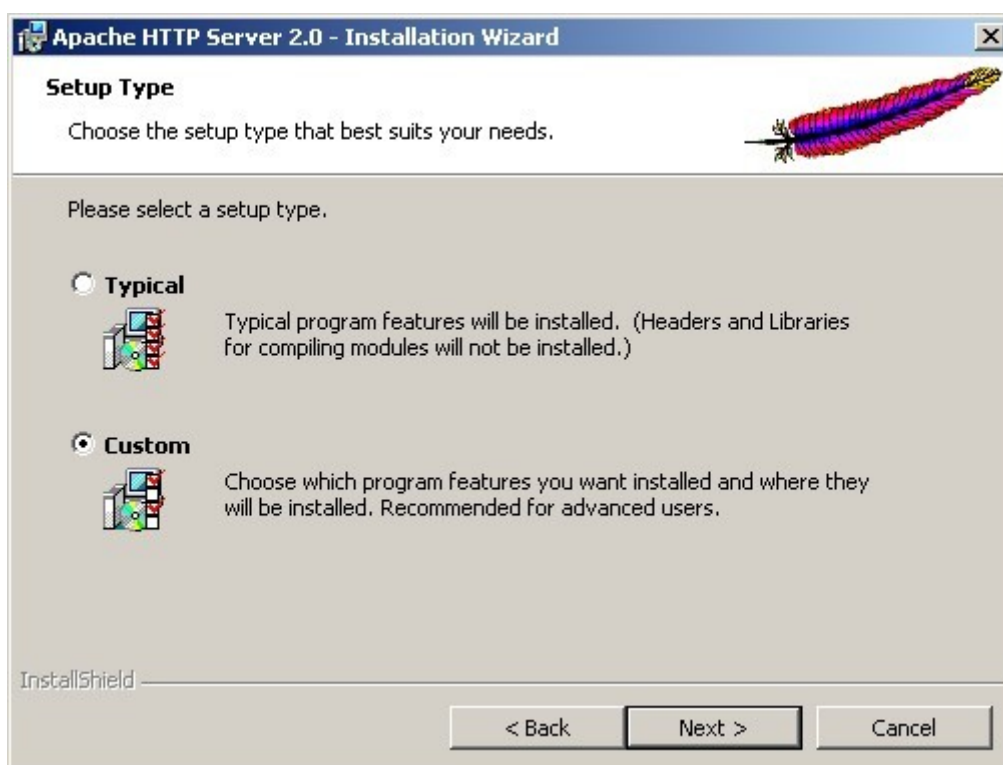


Illustration 6: Ecran de choix du type d'installation Typical ou Custom

On choisi le type d'installation pour notre serveur web :

- **Typical** : Installation typique (va installer les principaux éléments pour l'utilisation du serveur) – recommandé si vous n'avez pas trop d'idées des composants dont vous avez besoin.
- **Custom** : Installation personnalisée (on définit les composants que l'on souhaite installer pour notre serveur).

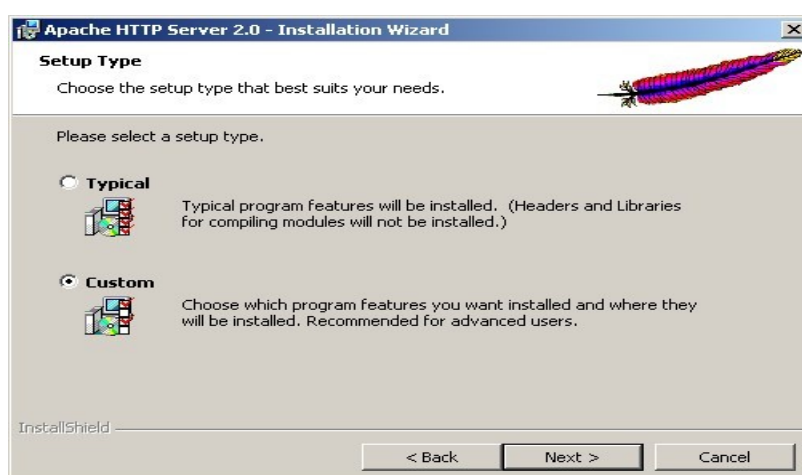


Illustration 7: Choix d'installation dans le cas présent : Custom

Dans, le cas présent, je choisis une installation de type **Custom**, et je clique sur **Next**.

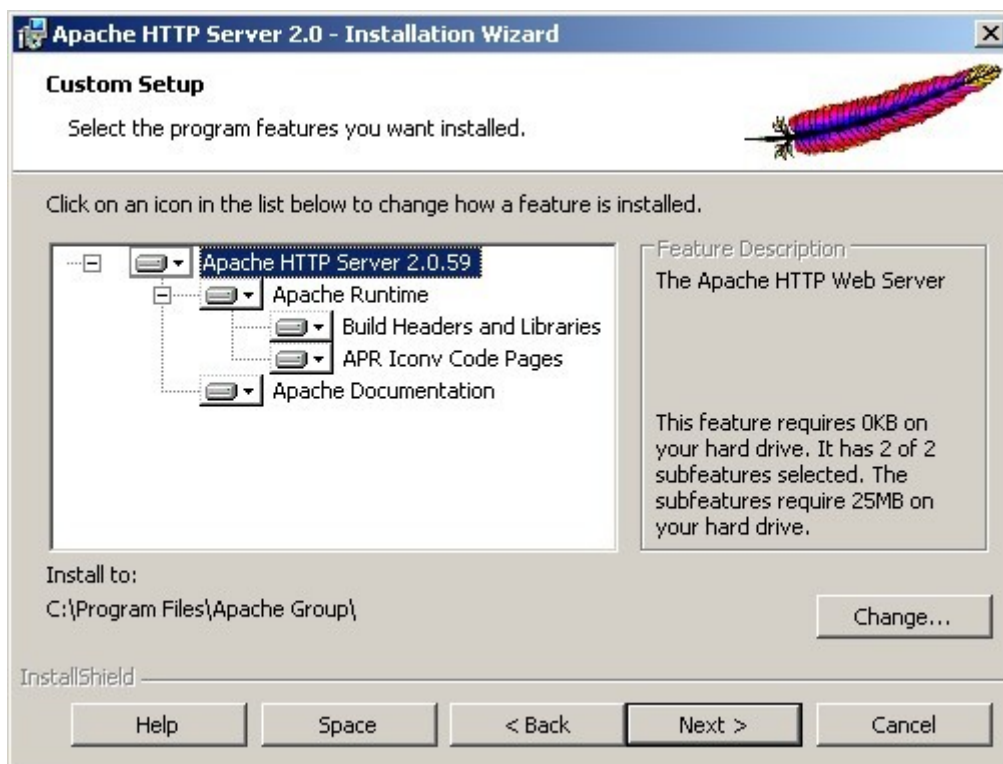


Illustration 8: Ecran de choix des composants à installer

Nous avons (ci-dessous) les principaux composants pour notre serveur web, à noter que tous ne sont pas utiles si l'on pense utiliser notre serveur de manière standard.

Dans le cas présent, on souhaite procéder à l'installation dans un répertoire autre que celui par défaut « **C:\Program Files\Apache Group** ».

Pour se faire, on clique sur le bouton **Change** pour obtenir l'écran suivant :

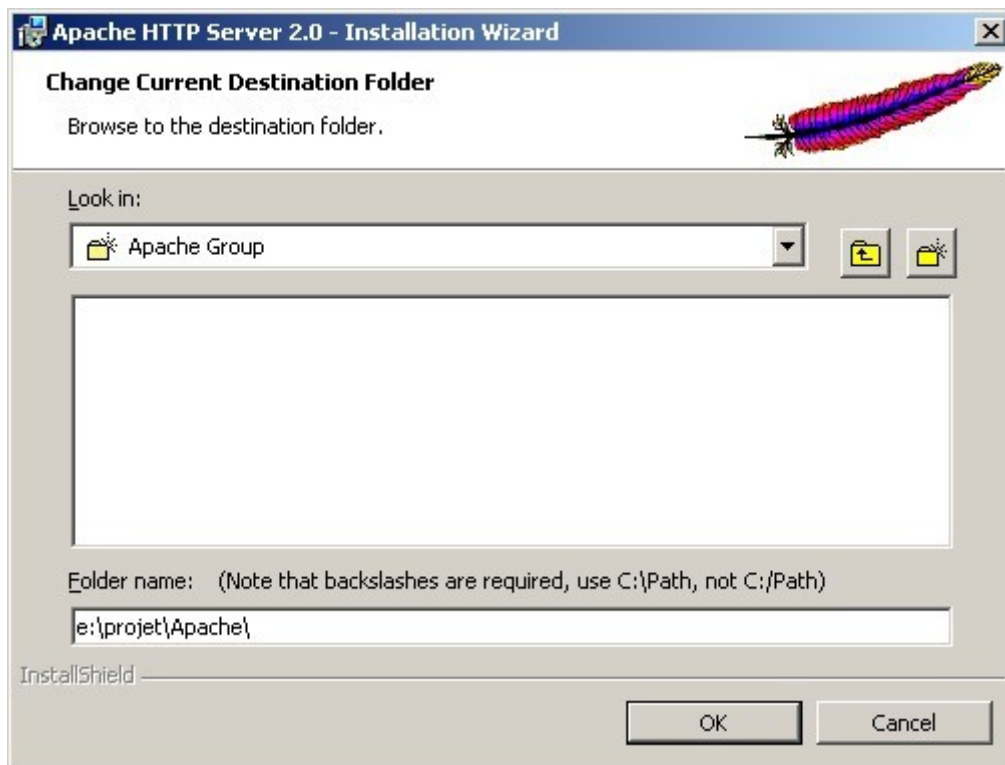


Illustration 9: Ecran choix de la destination pour l'installation d'Apache

Ici, Apache sera installé dans le répertoire « **E:\projet** ». On clique sur le bouton **OK** pour valider puis sur le bouton **Next** pour continuer notre installation.

Si l'on fait une erreur sur le disque de destination, le message suivant s'affichera :



Illustration 10: Ecran d'erreur si le disque de destination est invalide

Si tel est le cas, on vérifie le répertoire de destination ainsi que le lecteur sur lequel on souhaite faire l'installation.

On clique sur le bouton **Install** (illustration 11) pour procéder à l'installation.

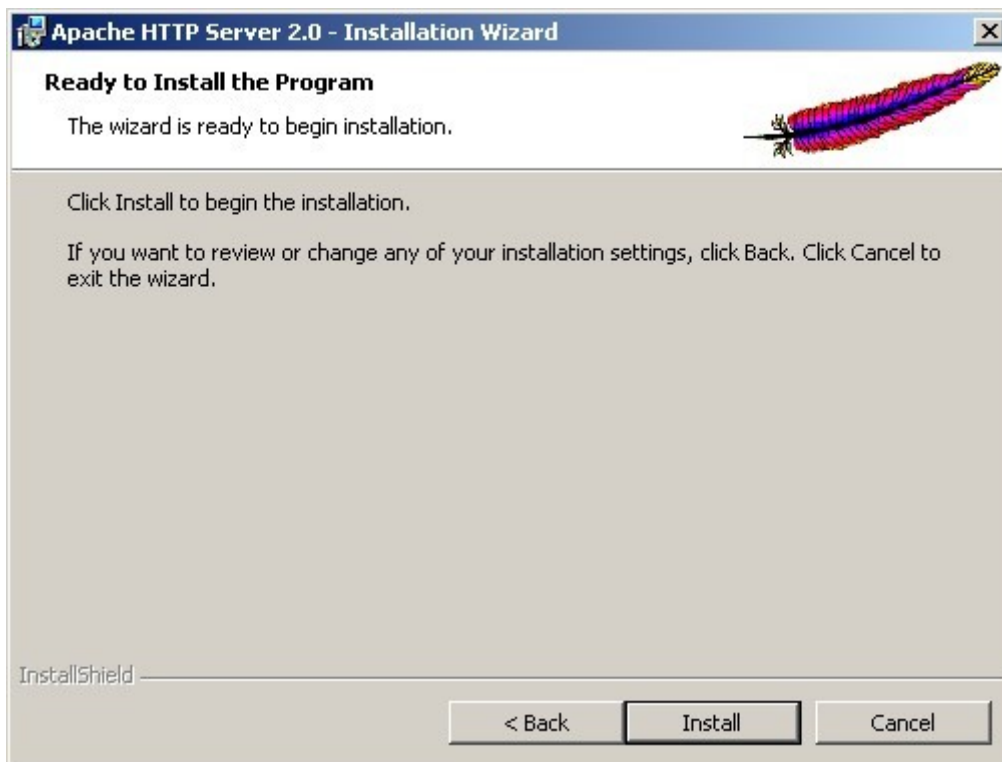


Illustration 11: Ecran pour lancer l'installation

Patience, installation en cours...

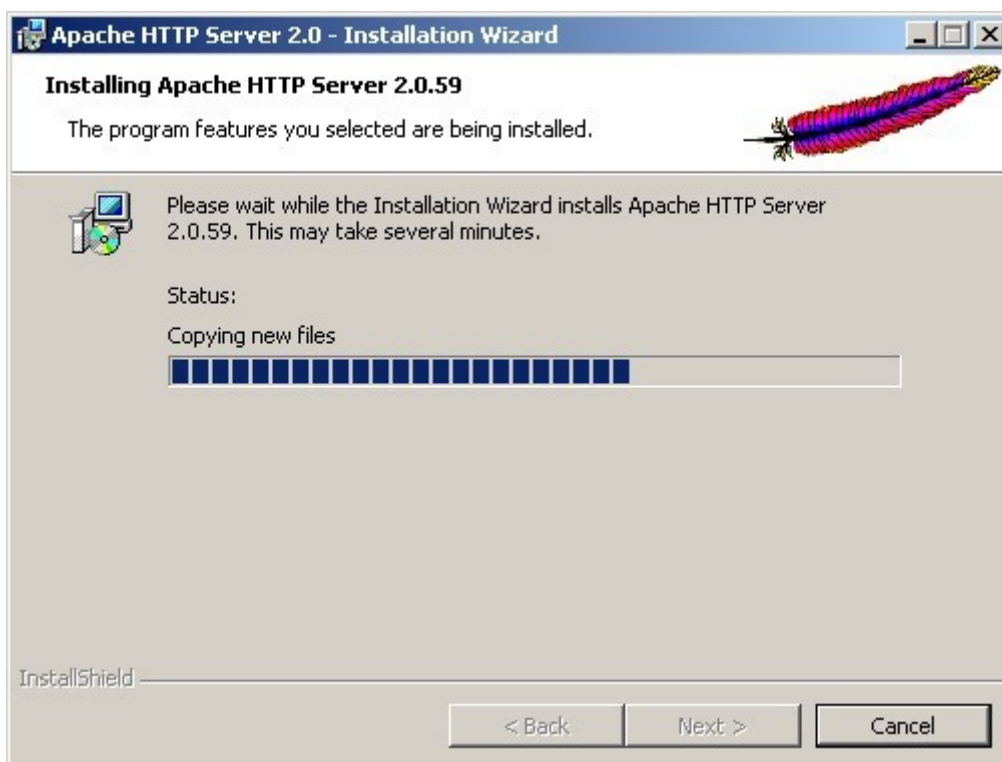


Illustration 12: Ecran de copie des fichiers

Si vous souhaitez annuler votre installation, cliquer sur le bouton **Cancel**.

Une fois l'installation terminée, nous avons l'écran ci-dessous qui s'affiche.

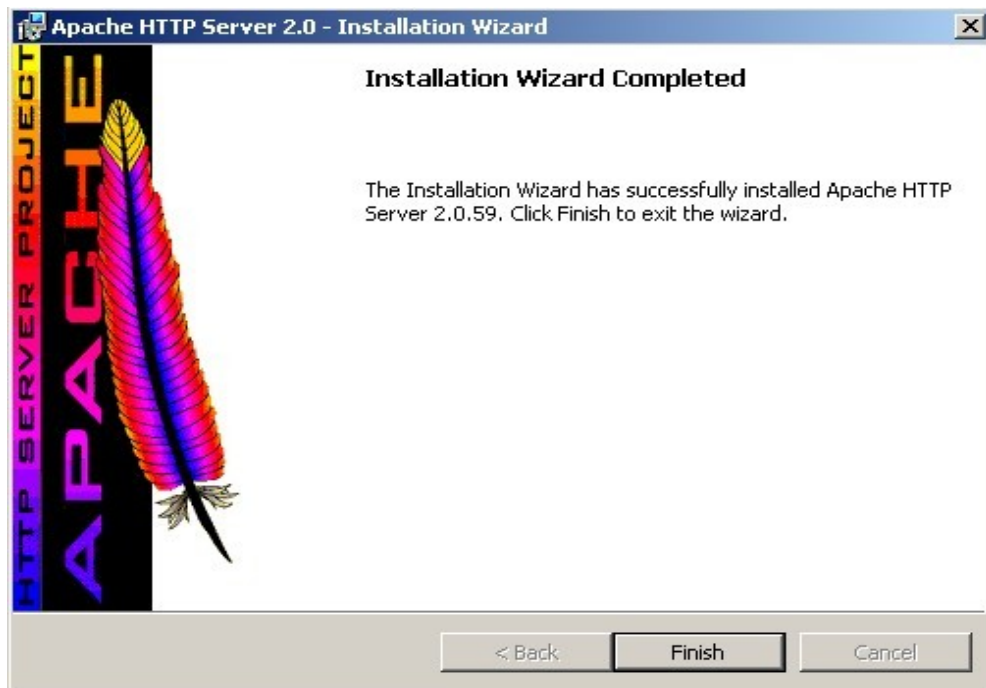


Illustration 13: Ecran de fin d'installation

On clique sur le bouton **Finish**.

NOTE : si vous utilisez le pare-feu intégré de **Windows XP service pack 2**, vous obtenez l'écran (ci-dessous), pour autoriser l'application Apache, cliquer sur le bouton **Débloquer**.

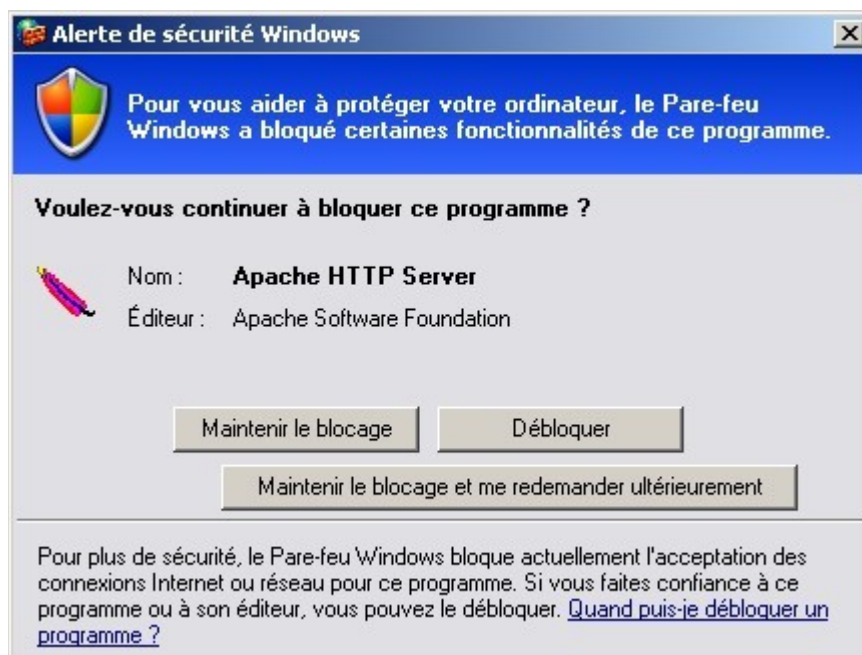


Illustration 14: Ecran du pare-feu windows qui propose de maintenir ou débloquent le programme

A présent, notre serveur est opérationnel, d'ailleurs, si l'installation était standard, les services sont déjà en action. Par ailleurs, si vous regardez dans le menu Démarrer / Programmes, vous devriez avoir le menu suivant :

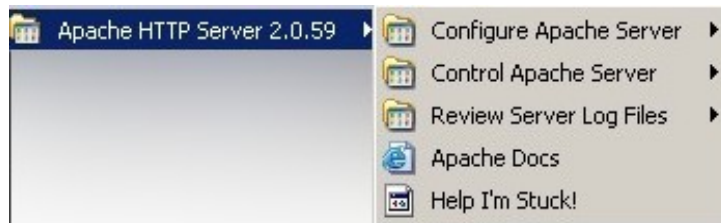



Illustration 15: Menu Apache dans le menu démarrer / Programmes

Vous aurez pu aussi remarquer la présence de **Apache Service Monitor** dans votre systray (à la gauche de l'heure) qui est représenté par l'icône ci-dessous :



Cet utilitaire vous permet de démarrer / arrêter / redémarrer le service Apache2. Pour accéder à Apache monitor, un clic droit sur  vous affichera le menu... **Open Apache Monitor**. Ce dernier vous permettant de lancer ou non votre serveur web Apache.

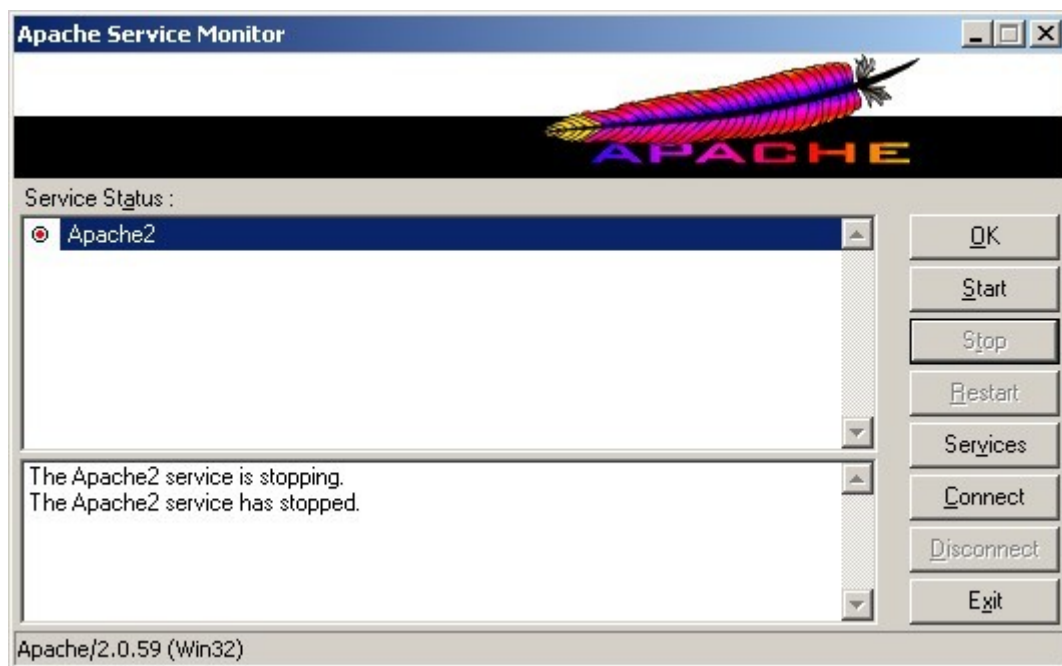


Illustration 16: Ecran de Apache Service Monitor disponible dans le systray

Note : Si vous souhaitez lancer votre service Apache sans pour cela passer par la fenêtre principale d'**Apache Service Monitor**. Il vous suffit de faire un **clic droit sur Open Apache Monitor**, ensuite choisissez l'option **Open Services**. Ceci aura pour effet d'afficher

la page des services. Là, on trouve le service **Apache2** sur lequel on va faire un clic droit, puis **démarrer** ou **arrêter** selon le cas.

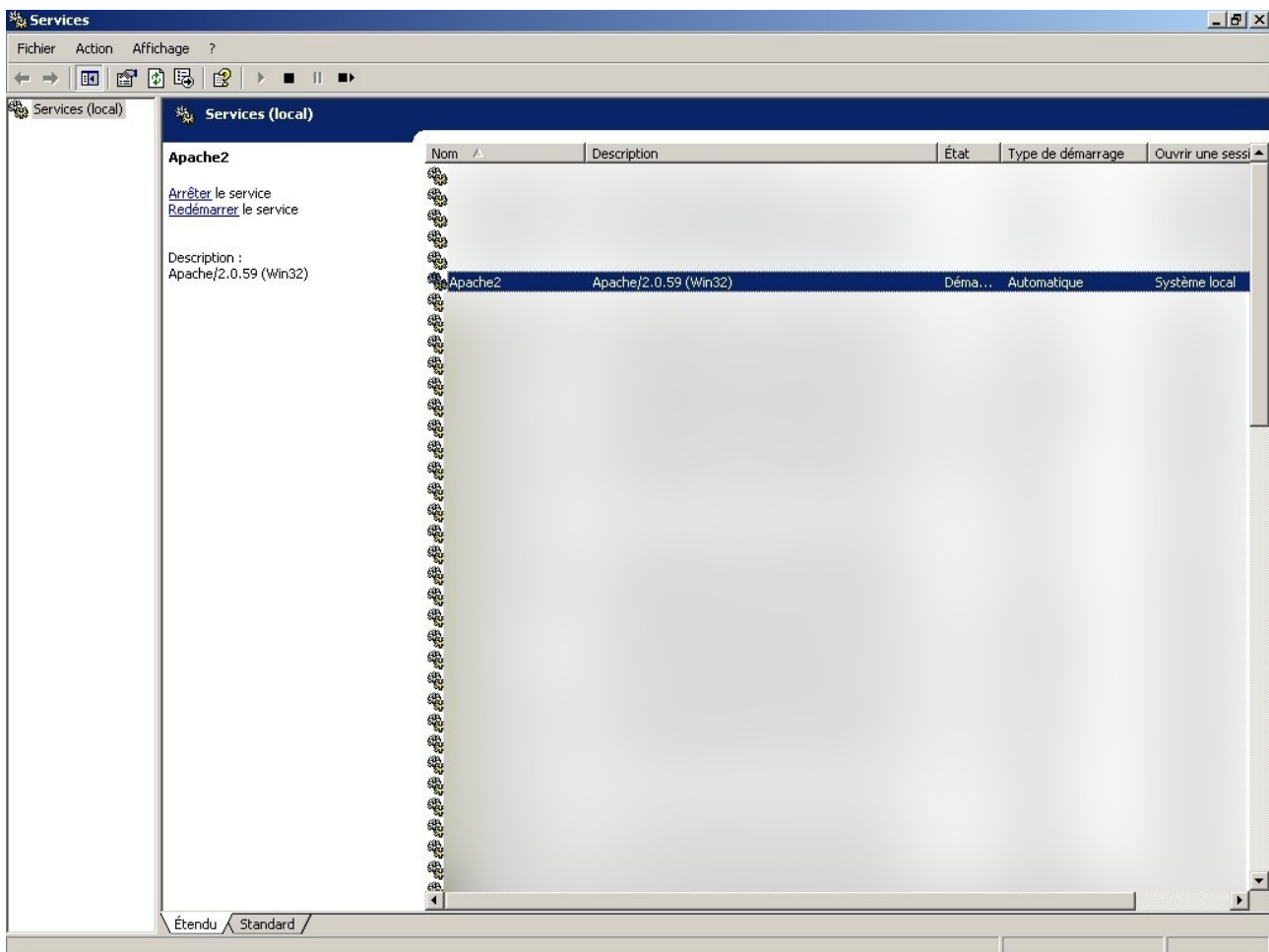


Illustration 17: Ecran des services de Windows, Apache2 est disponible dans ce dernier

Une astuce toute simple est d'ouvrir une invite de commande, pour cela, **Démarrer / Exécuter** puis taper **CMD** (ou cmd, ça n'a pas d'importance). La fenêtre **invite de commandes** s'ouvre, on tape :

NET START APACHE2 (pour démarrer le service)

NET STOP APACHE2 (pour arrêter le service)

Voilà, votre service sera lancé ou arrêté selon le cas.

Ceci étant nous allons effectuer une vérification pour savoir si notre serveur est bien en cours de fonctionnement.

Ma première visite !

Pour cela, on ouvre notre navigateur favori (Firefox par exemple), puis on tape dans la barre d'adresse :

<http://localhost> ou <http://127.0.0.1>

Si lors de l'installation vous avez choisi l'option « **only for the Current User, on Port 8080, when started Manually** », vous devez ajouter **:8080** à la fin de l'url comme ceci :

<http://localhost:8080> ou <http://127.0.0.1:8080>

Ceci n'étant valide que dans le cas où vous avez entré les mêmes informations indiquées lors de l'installation.

Si vous avez rentré une ip de votre réseau local en lieu et place de **127.0.0.1**. Votre serveur est visible à l'url suivante depuis n'importe quel poste de votre réseau [LAN](#).

<http://192.168.0.1> (par exemple)

Si tout a été bien installé et fonctionne, vous devriez avoir un écran similaire à celui-ci.

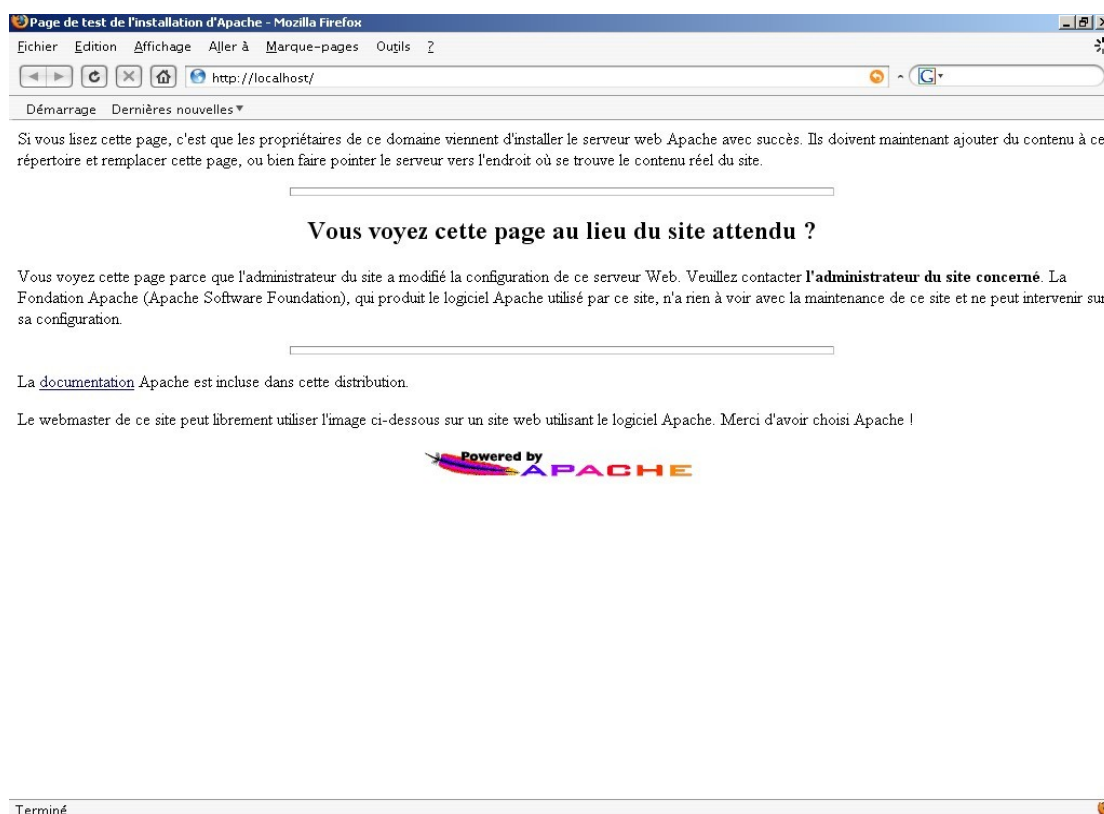


Illustration 18: Ecran de Firefox qui affiche la page par défaut d'Apache

Votre serveur web est à présent online.

En cas de problème :

Si pour une raison ou une autre, vous n'arrivez pas à accéder à la page par défaut de Apache (ci-dessus), il vous faut vérifier les points suivants :

Vérifier que le service Apache2 est bien démarré, pour cela clic droit sur le **poste de travail** puis **Gérer** et choisir la rubrique **Services et applications / Services** dans lequel vous aurez l'état actuel du service en question (dans le cas présente Apache2). Par ailleurs, vérifier le fichier log d'Apache, ce dernier se nomme **error.log** et se trouve dans le sous-répertoire **logs** de votre Apache. Celui-ci inscrivant les erreurs qui peuvent être intervenues lors du lancement du service ainsi que pendant son fonctionnement.

Skype et cie : Suite à un problème que j'ai rencontré récemment, vous prendrez soin lors de votre installation de ne pas avoir le logiciel **skype** en fonctionnement ainsi que tout autre logiciel qui pourraient utilisés le port **80** ou **8080** (selon l'installation choisie). Ce qui a pour effet, dans le cas présent lorsqu'on veut installer / démarrer le service Apache, d'afficher ce message d'erreur.

```
Apache2\bin>apache -k install
(OS 10048)Une seule utilisation de chaque adresse de socket (protocole/adresse r
éseau/port) est habituellement autorisée. : make_sock: could not bind to addre
ss 0.0.0.0:80
no listening sockets available, shutting down
Unable to open logs

Apache2\bin>net start apache
Le nom de service n'est pas valide.

Vous obtiendrez une aide supplémentaire en entrant NET HELPMSG 2185.
```

Illustration 19: Ecran invite de commandes qui affiche une erreur lors de l'installation du service Apache

Le voyant est **vert** quand le service Apache2 est démarré...

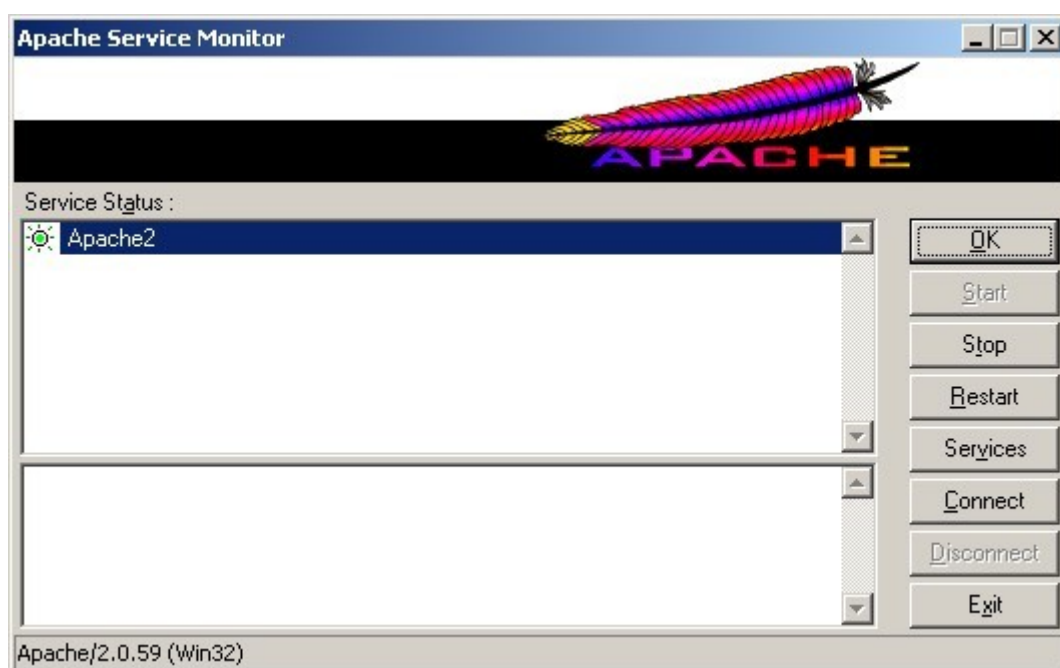


Illustration 20: Ecran Apache Service Moniteur avec service Apache lancé.

Configuration d'APACHE :

La configuration d'Apache se fait de manière simple via un seul et unique fichier de configuration. Le fichier de configuration du serveur web se nomme **httpd.conf** (un fichier texte qui sera édité avec le bloc-notes) et est situé dans le sous-répertoire **conf** d'Apache.

Ce fichier contient les principaux éléments pour faire en sorte que votre serveur web tourne sans encombre. Une modification dans ce fichier peut rendre indisponible Apache.

Voyons quelques paramètres modifiables sans trop de difficulté.

Tout d'abord, une chose bien utile si vous ne souhaitez pas utiliser le répertoire de base d'Apache pour vos documents web. Par défaut, le sous-répertoire qui contient les pages web se nomme **htdocs**, si vous souhaitez modifier cela, repérez le paramètre « **DocumentRoot** » puis modifier comme ceci :

DocumentRoot "e:/projet/www" (**On aura pris soin de créer le répertoire www**)

On aura pris soin de créer le sous-répertoire **www** dans **e:/projet** avant même d'avoir fait la modification dans le fichier de configuration, sinon cela aurait pour effet de générer une erreur lors du lancement d'Apache.

Si pour une raison ou pour un autre, on souhaite modifier l'adresse e-mail de l'administrateur du serveur, on repère le paramètre **ServerAdmin** puis on lui indique en valeur une adresse e-mail qui soit de préférence valide.

ServerAdmin contact@monserveurperso.com

Si l'on souhaite indiquer les fichiers qui seront traités comme des fichiers de base du serveur web, c'est-à-dire la page par défaut d'un répertoire web, nous pouvons modifier pour cela le paramètre **DirectoryIndex**.

DirectoryIndex index.htm index.html index.php index.php5

Ici, toutes les pages qui se nomment index.html, index.htm, index.php ou index.php5 seront prises en compte par le serveur web comme page par défaut d'un site web.

Pour faire en sorte que le visiteur est un minimum d'information concernant votre serveur lorsque une page d'erreur type 404 s'affiche, nous pouvons modifier la valeur du paramètre **ServerTokens**.

ServerTokens Prod

En donnant la valeur **Prod** cela permet de ne fournir que le nom du serveur, soit dans le cas présent **Apache**, il n'y aura aucune information concernant la version utilisée ni d'autres informations qui pourraient renseigner une personne mal intentionnée.

Par ailleurs, je veille à fournir une adresse e-mail qui pourrait permettre au visiteur de m'informer d'un éventuel problème sur le serveur. Pour cela je modifie la valeur du paramètre **ServerSignature**.

Comme ceci :

ServerSignature Email

Ce qui au final lorsqu' un message d'erreur est affiché permet à tout visiteur de pouvoir prévenir l'administrateur du serveur. On prendra soin de mettre une adresse e-mail valide.

Une option qui est très utile est l'utilisation du module status. Un module est une fonction qui permet d'ajouter des fonctions à votre serveur web.

Le module **status** permet dans le cas présent d'obtenir des informations en quasi temps réel sur l'état du serveur.

Pour ce faire dans le fichier de configuration, je vais rechercher la ligne suivante :

```
# LoadModule status_module modules/mod_status.so
```

Dans le cas présent, la ligne est actuellement en commentaire puisque ayant un # en son début de ligne.

Donc, on décommente tout d'abord la ligne :

```
LoadModule status_module modules/mod_status.so
```

Puis l' on recherche les quelques lignes ci-dessous (lignes qui dans leur version d'origine sont là aussi commentés #) :

```
<Location /server-status>  
    SetHandler server-status  
    Order deny,allow  
    Deny from all  
    Allow from .localhost (on supprime le . devant localhost)  
</Location>
```

Ces quelques lignes permettent de rendre ou non disponible l'état du serveur. Le **Deny from all** permet tout d'abord **interdit l'accès à tout le monde** puis avec l'option **Allow from**

localhost de l'**autoriser uniquement à localhost** (la consultation sera donc possible que depuis le serveur et non depuis une machine dans le réseau local par exemple).

Une fois la modification effectuée, on enregistre le fichier de configuration. On redémarre le serveur web (clic gauche sur **Open Apache Monitor / Apache2 / Restart**), ceci fait, on ouvre son navigateur favori (Opéra par exemple) et on tape l'url suivante dans la barre d'adresse.

<http://localhost/server-status>

Un résultat (ci-dessous) comparable à celui-ci s'affichera :

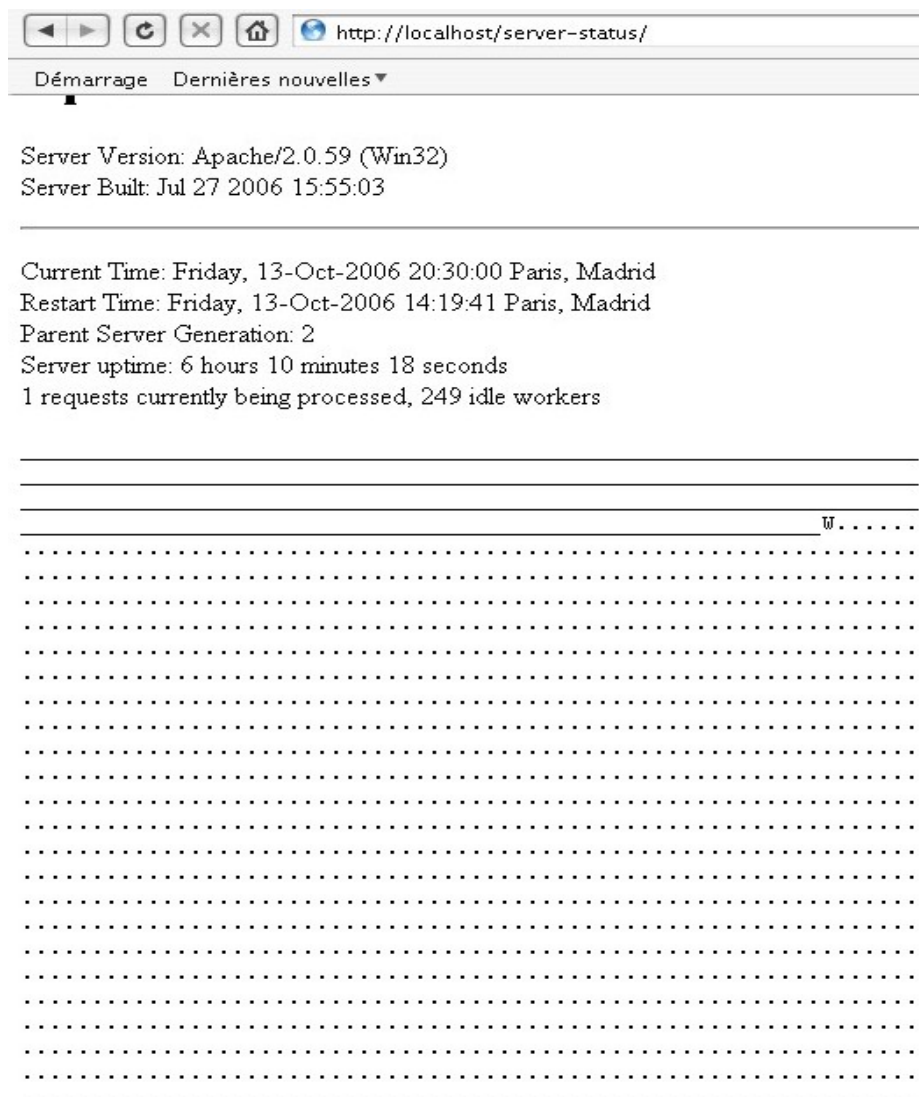


Illustration 21: Ecran de Firefox qui affiche la page server-status

Voilà donc en somme un aperçu de la configuration d'un serveur web Apache.

Note : Le module info (**mod_info.so**) permet dans le même esprit d'afficher la liste des modules en cours d'utilisation ainsi que les paramètres du serveur Apache.

Un mot sur les hôtes virtuels (Virtual Hosts) :

Qu'est-ce qu'un hôte virtuel ? Un hôte virtuel permet la création d'un sous-domaine au sein d'un domaine. Exemple : **www.monserveurperso.com** est un domaine qui a comme sous-domaine (entre autres) **forum.monserveurperso.com**.

Les hôtes virtuels sont mis en pratique lorsque l'on souhaite héberger de multiples sites sur son serveur web. Le principe étant que pour un sous-domaine on indique le paramètre correspondant au répertoire qui sera lié à ce dernier, ainsi que divers autres paramètres plus ou moins utiles selon les besoins.

Voici comment se présente la section concernant les **Virtual Hosts** dans le fichier de configuration Apache. Comme vous pouvez le remarquer tout y est commenté.

```
#NameVirtualHost *:80
```

```
#<VirtualHost *:80>
```

```
# ServerAdmin webmaster@dummy-host.example.com
```

```
# DocumentRoot /www/docs/dummy-host.example.com
```

```
# ServerName dummy-host.example.com
```

```
# ErrorLog logs/dummy-host.example.com-error_log
```

```
# CustomLog logs/dummy-host.example.com-access_log common
```

```
#</VirtualHost>
```

Pour illustrer le propos, nous allons créer un hôte virtuel pour le sous-domaine **demo** qui fera partie du domaine **monserveurperso.com**.

```
NameVirtualHost *:80
```

```
<VirtualHost *:80>
```

```
ServerAdmin webmaster@monserveurperso.com (une adresse valide de préférence)
```

```
DocumentRoot e:/projet/www/demo/
```

```
ServerName demo.monserveurperso.com
```

```
ErrorLog logs/dummy-host.example.com-error_log
```

```
CustomLog logs/dummy-host.example.com-access_log common
```

```
Php_admin_value open_basedir "e:/projet/www/demo/" (à utiliser avec php)
```

```
</VirtualHost>
```

DocumentRoot permet d'indiquer le répertoire sur lequel pointera le sous-domaine **demo**. Dans le cas présent, le répertoire en question sera **e:/projet/www/demo**.

Les lignes **ErrorLog** et **CustomLog** ne sont nullement obligatoires, mais dans bien des cas ça permet de garder une trace des erreurs liées au sous-domaine en question.

Une fois les changements dans le fichier de config d' Apache enregistrés, il faut penser à redémarrer le serveur web. Par ailleurs n'oubliez pas de déclarer votre sous-domaine au niveau de votre DNS.

Si la personne tape **http://demo.monserveurperso.com** dans son navigateur, elle devrait obtenir le sous-domaine en question (une page web par exemple).

DANS LE CAS PRESENT LE SOUS-DOMAIN DEMO EST DE NATURE FICTIF PUISQUE N'ETANT PAS UTILISER SUR MONSERVEURPERSO.COM.

Si vous souhaitez de plus amples renseignements sur les Virtual Hosts je vous engage à la lecture de la documentation officielle sur le sujet disponible à l'url suivante :

<http://httpd.apache.org/docs/2.0/vhosts/>

Ceci étant, la configuration du serveur web Apache étant d'une grande richesse, je vous invite à lire **le manuel de référence** pour la version **2.0** qui se trouve dans le sous-répertoire **manual** d'Apache ou sur sa version online disponible à l'url suivante :

<http://httpd.apache.org/docs/2.0/>

Qu'est-ce que PHP ?

Notre serveur web Apache installé, nous allons procéder à l'ajout du support de ce dernier pour le langage PHP. Avant toute chose, voyons ce qu'est PHP.

Pour cela voyons ce que nous dit [wikipedia](#) sur le sujet.

PHP est l'acronyme récursif de PHP: **Hypertext Preprocessor**.

PHP est un langage de script qui est principalement utilisé pour être exécuté par un serveur HTTP, mais il peut fonctionner comme n'importe quel langage interprété en utilisant les scripts et son interpréteur sur un ordinateur. PHP permet de développer des scripts suivant le modèle procédural et/ou le modèle objet. On désigne parfois PHP comme une plate-forme plus qu'un simple langage.

NDLA : En d'autres termes cela nous permet de bénéficier du support PHP sur notre serveur web qui sera utile si l'on souhaite mettre en place un site dit dynamique.

Si vous souhaitez lire la définition complète concernant PHP, voyez la page wikipedia qui lui est consacrée : http://fr.wikipedia.org/wiki/PHP_hypertext_preprocessor

Téléchargement de PHP :

Avant toute chose, il nous faut aller sur le site officiel de [PHP.net](#), de là, nous allons sur la page *downloads*. Cette page va nous permettre de récupérer l'archive que l'on souhaite. A noter qu'à l'heure où j'écris ce document, la version disponible est la **5.1.6** (en date du 14 octobre 2006).

Les archives disponibles sont soit l'archive contenant un installer (**PHP 5.1.6 installer**), soit l'archive compressée (**PHP 5.1.6 zip package**). Personnellement, j'ai toujours procédé à l'installation de PHP en passant par l'archive zip. Donc, dans le cas présent, on prendra soin de récupérer l'archive zip (le fichier PHP 5.1.6 zip package) qui présente une taille de 9 Mo.

<http://fr.php.net/get/php-5.1.6-Win32.zip/from/a/mirror>

NDLA : l'archive PECL (**Collection of PECL modules for PHP 5.1.1**) est en fait une collection d'extensions pour PHP. Si vous souhaitez plus d'information consulter la page ci-après : <http://pecl.php.net/>

L'archive récupérée nous pouvons à présent passer à son installation...

Installation de PHP :

Localisez l'archive que vous avez téléchargée, par défaut sur le bureau si vous avez téléchargé avec Mozilla Firefox sinon une petite recherche (sur votre disque dur) vous permettra de localiser le programme d'installation de PHP.

L'archive se nomme dans le cas présent : **php-5.1.6-Win32.zip**

Nous allons passer à la décompression du fichier en question, pour cela, ouvrez votre programme de décompression favori (exemple : 7zip).

Contenu de l'archive vu par 7-zip :

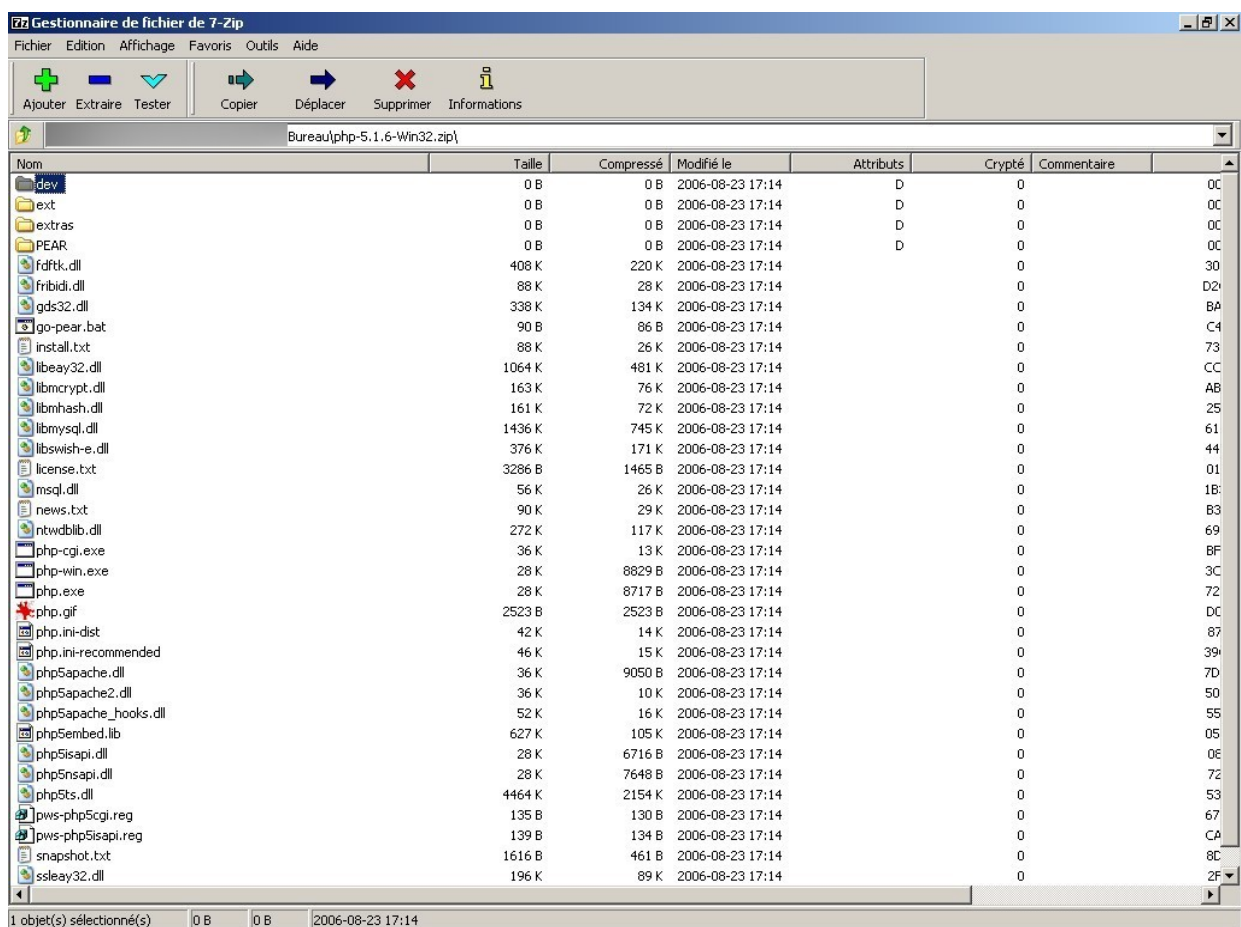


Illustration 22: Ecran de 7-zip qui affiche le contenu de l'archive

Nous allons décompresser cela dans le répertoire **e:\projet\php5**, là où nous avons installé notre serveur web.

Bien entendu, vous n'êtes en rien obligés de dézipper à cet endroit précis.

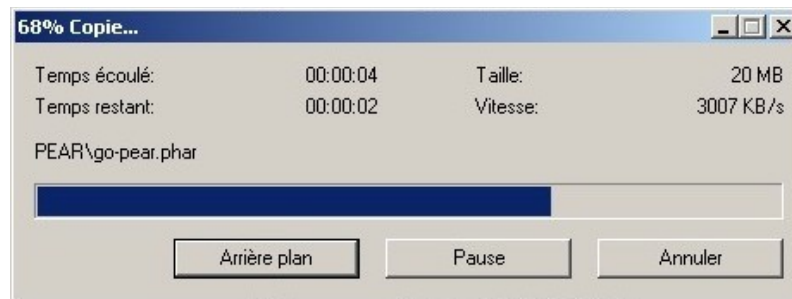


Illustration 23: Ecran de 7-zip qui affiche l'extraction des données de l'archive.

Extraire vers **e:\projet\php5**

Ceci fait, voyons les deux installations possibles de php pour le faire travailler avec Apache.

Soit installer PHP en tant que **module d'Apache**

Soit installer en tant que **programme CGI**

Dans le cas présent, je vais procéder à l'installation en tant que **module** de mon serveur Apache. Si vous avez jeter un oeil dans le fichier **install.txt** qui se trouve à la racine du répertoire php5, vous aurez vu que cela n'est pas très différent au niveau de la modification du fichier de configuration Apache (httpd.conf).

Ainsi, je vais devoir configurer Apache en conséquence.

Ceci étant, je dois éditer le fichier de configuration d'Apache qui se situe dans **e:\projet\Apache2\conf** et qui se nomme **httpd.conf**.

A celui-ci, je dois lui ajouter les lignes suivantes en respectant la place de chacune.

```
LoadModule php5_module "e:/projet/php5/php5apache2.dll"  
AddType application/x-httpd-php .php  
PHPIniDir "e:/projet/php5"
```

La première ligne (**LoadModule php5_module "e:/projet/php5/php5apache2.dll"**) s'ajoute à la suite des autres **LoadModule** que vous trouverez dans le fichier de configuration d'Apache.

Pour ce qui concerne la deuxième ligne (**AddType application/x-httpd-php .php**), elle s'ajoute à la suite des autres **AddType**. La troisième ligne (**PHPIniDir "e:/projet/php5"**) indique le répertoire de base de PHP, elle s'ajoute en dessous de la ligne **AddType application/x-httpd-php .php**. Ceci fait, vous enregistrez le fichier de configuration ainsi modifié.

Nous allons à présent passer à la configuration du **php.ini** qui à l'origine se trouve dans le répertoire d'installation sous le nom de **php.ini-dist**, première chose, en faire une copie et renommez la dite copie en tant que **php.ini**.

On édite le fichier **php.ini** pour modifier les lignes suivantes :

extension_dir = "./" en extension_dir = "e:/projet/php5/ext"

;upload_tmp_dir = en upload_tmp_dir = e:/projet/php5/uploadtemp

;session.save_path = "/tmp" en session.save_path = "e:/projet/php5/sessionsave"

;extension=php_mbstring.dll en extension=php_mbstring.dll

A noter que les deux dernières lignes ne sont nullement obligatoires, si l'on ne pense pas utiliser les fonctions qui font appel à ces valeurs. Dans un autre cas, on aura pris soin de créer les sous-répertoires **uploadtemp** et **sessionsave**.

Autre point, si vous pensez utiliser le serveur de données MySQL, il vous sera nécessaire de décommenter la ligne suivante :

;extension=php_mysql.dll en extension=php_mysql.dll

Par ailleurs, vous aurez pris soin de copier le fichier **libmysql.dll** dans le répertoire système soit **c:\windows\system32** ou **c:\winnt\system32** selon le système d'exploitation utilisé.

NDLA : Par défaut, **MySQL n'est plus activé dans PHP5** ce qui explique la manipulation vue ci-dessus. Si vous obtenez un message similaire à celui-ci :

"Unable to load dynamic library './php_mysql.dll'"

C'est tout simplement parce que le fichier **libmysql.dll** n'a pu être trouvé par le système.

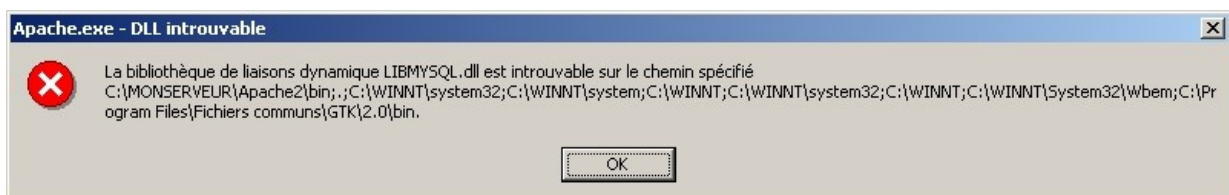


Illustration 24: Ecran d'erreur lié au fichier libmysql.dll

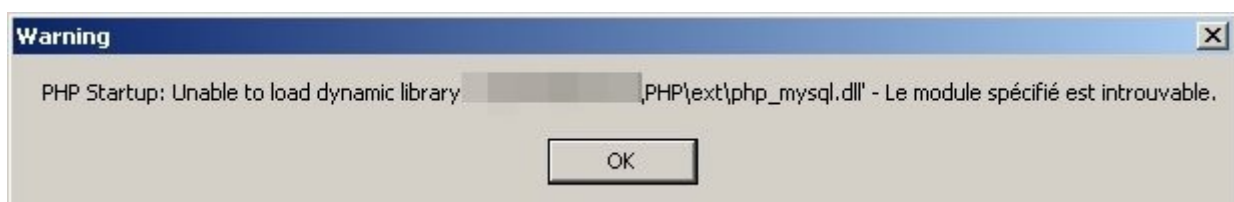


Illustration 25: Ecran d'erreur lié au fichier php_mysql.dll

Ceci étant nous pouvons à présent, faire un premier test pour savoir si notre serveur Apache a bien pris en compte le support PHP. Pour se faire, nous allons créer un fichier **info.php** que nous placerons à la racine de notre serveur web.


Dans le cas présent, notre racine se situe dans le répertoire suivant **e:\projet\www** (répertoire que l'on a spécifié lors de la configuration d'Apache).

Le fichier info.php contiendra la ligne suivante :

<? phpinfo(); ?>

La ligne ajoutée et le fichier modifié / enregistrer, ouvrez votre navigateur web favori et rendez-vous à l'url suivante :

http://127.0.0.1/info.php ou http://localhost/info.php

PHP Version 5.1.6

System	Windows NT	5.1 build 2600
Build Date	Aug 23 2006 16:31:18	
Configure Command	cscript/nologo configure.js "--enable-snapshot-build" "--with-gd=shared"	
Server API	Apache 2.0 Handler	
Virtual Directory Support	enabled	
Configuration File (php.ini) Path	C:\WINDOWS	
PHP API	20041225	
PHP Extension	20050922	
Zend Extension	220051025	
Debug Build	no	
Thread Safety	enabled	
Zend Memory Manager	enabled	
IPv6 Support	enabled	
Registered PHP Streams	php, file, http, ftp, compress.zlib	
Registered Stream Socket Transports	tcp, udp	
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, zlib.*	

This program makes use of the Zend Scripting Language Engine:
Zend Engine v2.1.0, Copyright (c) 1998-2006 Zend Technologies

**Powered By**

Illustration 26: Ecran de Firefox qui affiche le résultat de info.php

Cette page confirme la bonne prise en compte de PHP par Apache.

Un mot sur la sécurité :

Si vous souhaitez avoir un minimum de sécurité sur votre serveur web, il serait bien venu d'activer dans le fichier de configuration php l'option **safe_mod**. Cette option permet de mettre en place un contrôle au niveau des permissions d'accès aux fichiers ainsi que la limitation ou la désactivation de certaines fonctions.

safe_mode = Off en **safe_mod = On**

ATTENTION : certaines applications web ne fonctionnent pas normalement si le **safe_mod** est activé.

Ceci permet d'accroître la sécurité d'un serveur qui pourrait accueillir l'hébergement de sites.

Open_basedir : cette option permet de limiter l'accès aux fichiers contenu dans le (ou les) répertoire(s) spécifié(s). (Utilisable dans la configuration d'un hôte virtuel.)

Open_basedir = e:/projet/www/

Autre point, en phase développement de site ou d'application, il est bon de savoir d'où viennent les problèmes, par contre en phase de production (serveur accessible aux personnes), il n'est pas souhaitable de voir afficher les messages d'erreurs. Ainsi on utilisera l'option **display-errors**. Par défaut l'option est activée. Dans le cas présent nous souhaitons la désactiver :

display-errors = on en **display-errors = off**

Ceci étant, nous souhaitons pouvoir être informés des éventuelles erreurs de script php. Donc, on activera l'option **log_errors** et on définira le répertoire / fichier où sera sauvegarder le fichier de log.

log_errors = off en **log_errors = on**

et

;error_log = filename en **error_log = e:/projet/log/**

Si l'on souhaite avoir les erreurs dans les journaux d'événements, dans ce cas, on décommente la ligne **;error_log = syslog**.

Nous avons vu ici quelques options disponibles pour accroître la sécurité du serveur web que vous souhaitez mettre en place. Ceci étant cette liste n'est nullement exhaustive; il est recommandé la lecture de documents spécifiques à la sécurité de PHP ainsi que d'Apache pour parvenir à une sécurité optimale. Vous trouverez dans le chapitre Ressources des liens concernant la sécurité.

PHP est à présent installé sur notre serveur web. Je vous conseille par ailleurs la lecture du manuel PHP (en français) qui est disponible à l'url suivante :

<http://fr.php.net/manual/fr/>

Passons maintenant à l'installation de notre serveur de base de données, MySQL en l'occurrence.

Qu'est-ce qu'un serveur mysql ?

MySQL (SQL veut dire Structured Query Language en anglais) est un serveur de bases de données relationnelles SQL, très rapide, multithread, robuste et multi-utilisateurs. MySQL est un logiciel libre développé sous licence GNU General Public License. Il a été conçu et est maintenu par une société suédoise, MySQL AB, fondée par deux Suédois et un Finlandais : David Axmark, Allan Larsson et Michael « Monty » Widenius.

Au début, MySQL n'autorisait pas les transactions, mais cette fonction est maintenant disponible. Le standard ANSI SQL a quelquefois été étendu, mais n'est pas complètement implémenté, les fonctionnalités les plus souvent demandées sont sans doute :

Les « sous-sélections » ou « SELECTs imbriqués », maintenant disponibles, depuis la version 4.1, le codage sur plusieurs octets des caractères, comme UTF-8, est maintenant disponible.

MySQL fonctionne sur beaucoup de plates-formes différentes, incluant AIX, BSDi, FreeBSD, HP-UX, Linux, Mac OS X, NetBSD, OpenBSD, OS/2 Warp, SGI Irix, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64 Unix, Windows 95, 98, NT, 2000 et XP.

Les bases de données MySQL sont accessibles en utilisant les langages de programmation C, C++, Eiffel, Java, Perl, PHP, Python, et Tcl. Une API (interface de programmation) spécifique est disponible pour chacun d'entre eux. Une interface ODBC appelée MyODBC est aussi disponible.

Les moteurs intégrés inclus dans MySQL sont nombreux, entre autres : MyISAM, MERGE, ISAM, InnoDB, Berkeley DB, etc...

La question ayant obtenue une réponse... Nous allons procéder maintenant à l'installation.

Téléchargement de MySQL :

Avant l'installation, nous devons tout d'abord récupérer l'archive contenant notre futur serveur mysql. Rendez-vous sur le site mysql.com ou sur fr.mysql.com. De là, il faut récupérer la version **5.0.XX** de MySQL pour environnement MS Windows.

Attention, il y a **3** versions différentes de l'archive :

- **Windows Essentials**
- **Windows**
- **Without installer (unzip in C:\)**

Dans le cas présent, nous allons baser notre installation sur MySQL version **5.0.27** en essentials, version qui est actuellement diffusée à l'heure à laquelle j'écris ce document.

Nom du fichier téléchargé : **mysql-essential-5.0.27-win32.msi**

Bien entendu, il va de soi que vous pouvez baser votre installation sur une version supérieure. Vous pouvez effectuer le téléchargement de l'application à cette url :

<http://dev.mysql.com/get/Downloads/MySQL-5.0/mysql-essential-5.0.27-win32.msi/from/ftp://ftp.inria.fr/pub/MySQL/>

Ceci fait, nous pouvons à présent installer MySQL sur notre machine.

Installation de MySQL :

Localisez l'archive que vous avez téléchargée, par défaut sur le bureau si vous avez téléchargé avec Firefox sinon une petite recherche (sur votre disque dur) vous permettra de localiser le programme d'installation de MySQL.

*Un double-clic lancera le programme d'installation... On clique sur **Next**.*



Illustration 27: Ecran d'introduction de l'installation de MySQL

On choisit le type d'installation, par défaut, c'est une installation **Typical**. Dans le cas présent, je vais faire une installation personnalisée (**Custom**).



Illustration 28: Ecran de choix de type d'installation : Typical / Complete / Custom

On coche **Custom** puis on clique sur Next.

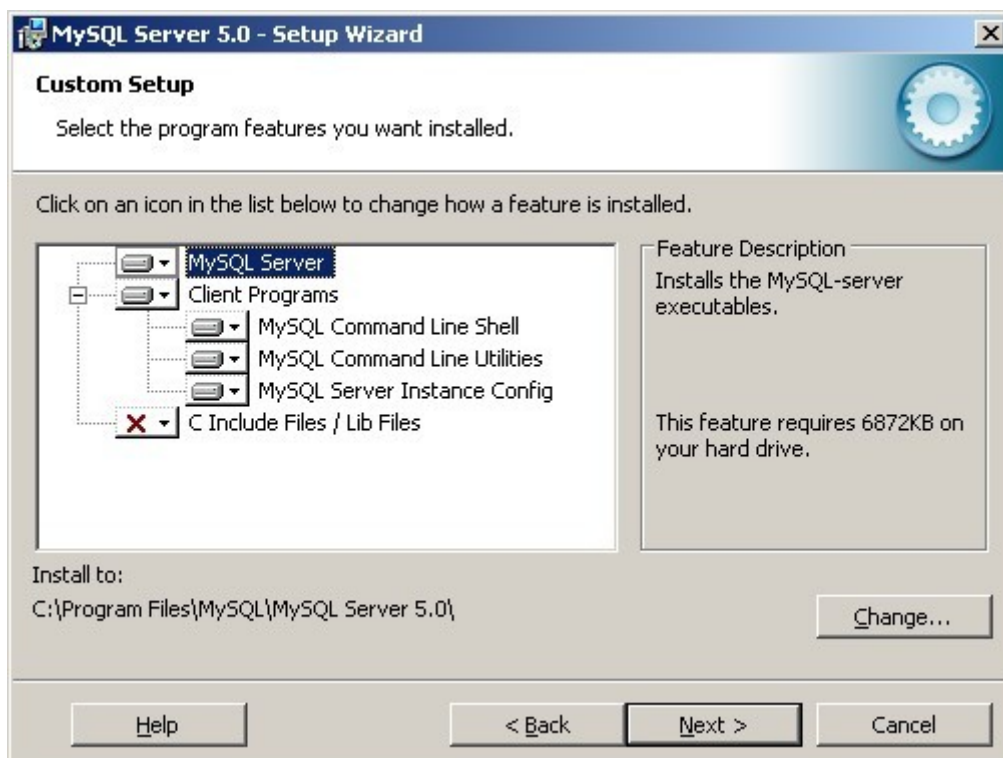


Illustration 29: Ecran du choix des composants à installer

Je détermine les composants que je souhaite installer, par ailleurs, je peux changer le répertoire où sera installé mon serveur MySQL.

Par défaut : **C:\Program Files\MySQL\MySQL Server 5.0**

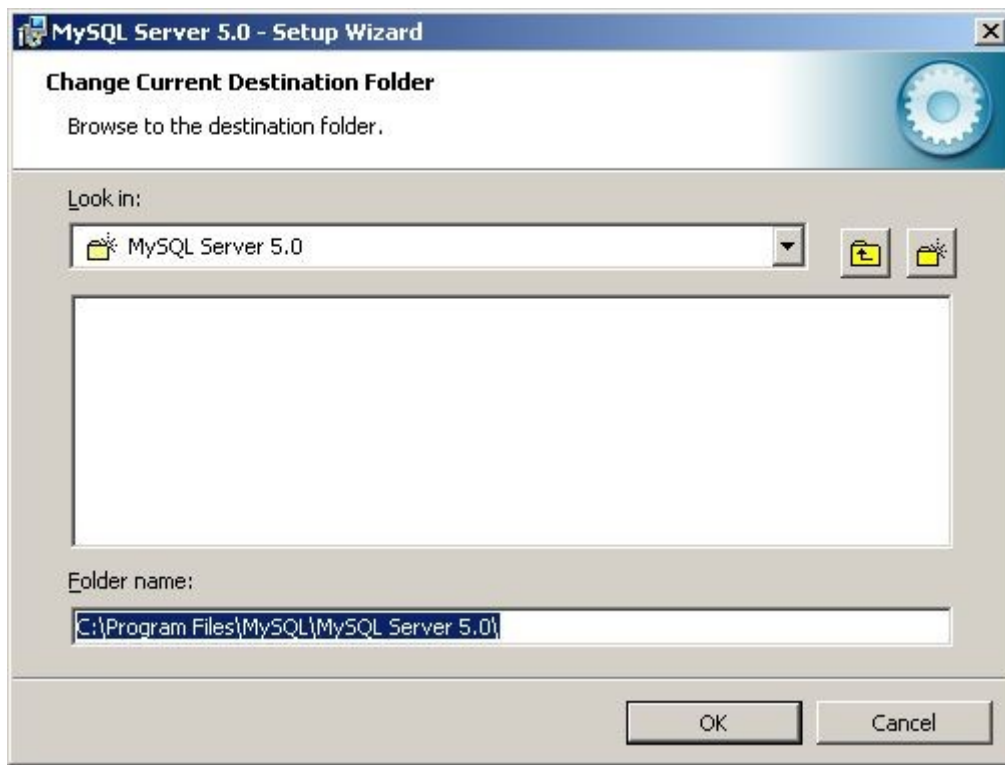


Illustration 30: Ecran choix du répertoire de destination

On clique sur le bouton **Change** si l'on souhaite changer le répertoire d'installation. Dans le cas présent mon serveur sera dans **E:\Projet\MySQL Server 5.0** Je peux cliquer sur bouton **Install**.

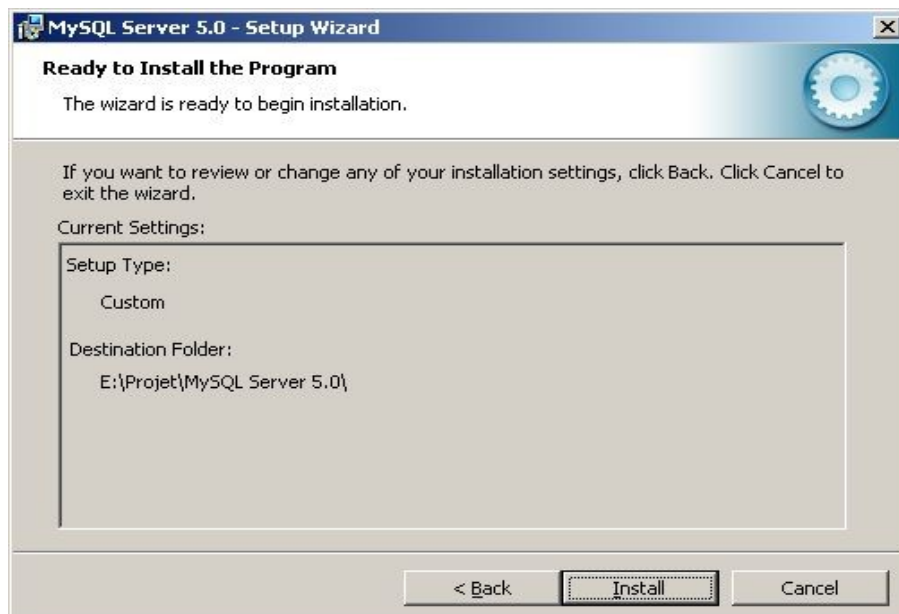


Illustration 31: Ecran récapitulatif avant installation de MySQL

L'installation est en cours, si je souhaite l'interrompre je clique sur le bouton **Cancel**.



Illustration 32: Ecran installation en cours

Une fois la copie des fichiers terminée, nous passons à la configuration de notre serveur MySQL.



Illustration 33: Ecran création d'un compte MySQL.com

Cet écran (ci-dessus), vous demande si vous souhaitez créer un compte ou si vous en disposez un sur le site officiel mysql.com. Dans le cas présent, je ne souhaite ni créer un compte ni m'authentifier sur le site. Donc, je coche **Skip Sign-Up** (ci-dessous).



Illustration 34: Ecran création du compte. Option Skip Sign-Up choisie

Dans le cas présent, je souhaite procéder à la configuration de mon serveur, je coche l'option **Configure the MySQL Server Now**.



Illustration 35: Ecran qui demande si l'on veut configurer son serveur MySQL
On clique sur le bouton **Finish**.

Configuration de MySQL :

Nous avons choisi précédemment (Illustration 35) l'option **Configure the MySQL Server now**, ceci va nous permettre de configurer le serveur MySQL dès à présent.



Illustration 36: Ecran d'introduction configuration de MySQL

On clique sur le bouton **Next** pour débiter la configuration.

Dans le cas présent, on choisit une configuration de type **Standard**, donc on coche l'option **Standard Configuration**.

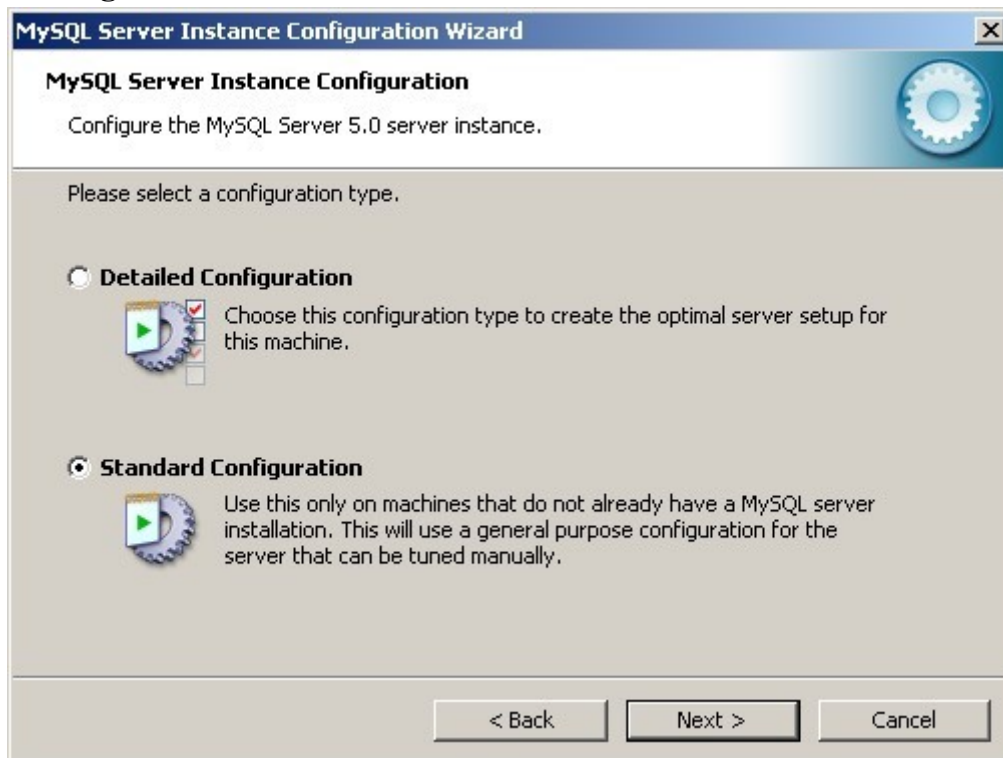


Illustration 37: Ecran type de configuration : Detailed Configuration / Standard Configuration

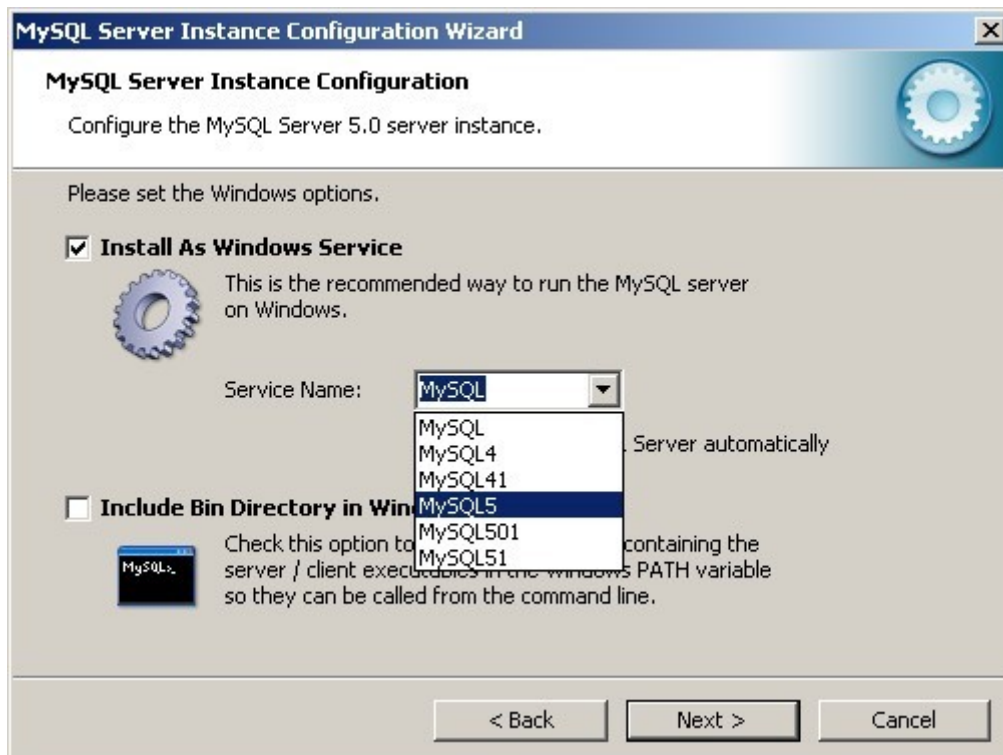


Illustration 38: Ecran installation du service et définition du nom du service ajout d'un path dans les variables d'environnement de Windows

Ici, nous pouvons choisir le nom du service (**Service Name**) qui va être installé, dans le cas présent, **MySQL5**. Par ailleurs, on peut ajouter une variable **Path** dans la configuration de Windows ce qui aura pour effet de rendre disponibles les commandes MySQL depuis n'importe quel répertoire dans une invite de commandes.



Illustration 39: Ecran définition du nom du service et lancement automatique du service



Illustration 40: Ecran définition du mot de passe root / création d'un compte anonyme

Nous devons déterminer le mot de passe du compte administrateur (**root**) du serveur. Par ailleurs, nous pouvons procéder à la création d'un utilisateur anonyme sur le serveur. Ceci étant, dans le cas présent, nous définissons le mot de passe root mais nous ne créons pas d'utilisateur anonyme qui comme on peut le voir n'est pas des plus prudent (**insecure**) pour le serveur.

NOTE : Je ne peux que vous conseiller de mettre un mot de passe alpha-numérique pour assurer un bon niveau de sécurité sur votre serveur MySQL.

Nos options de configuration définies, MySQL va procéder à son exécution.

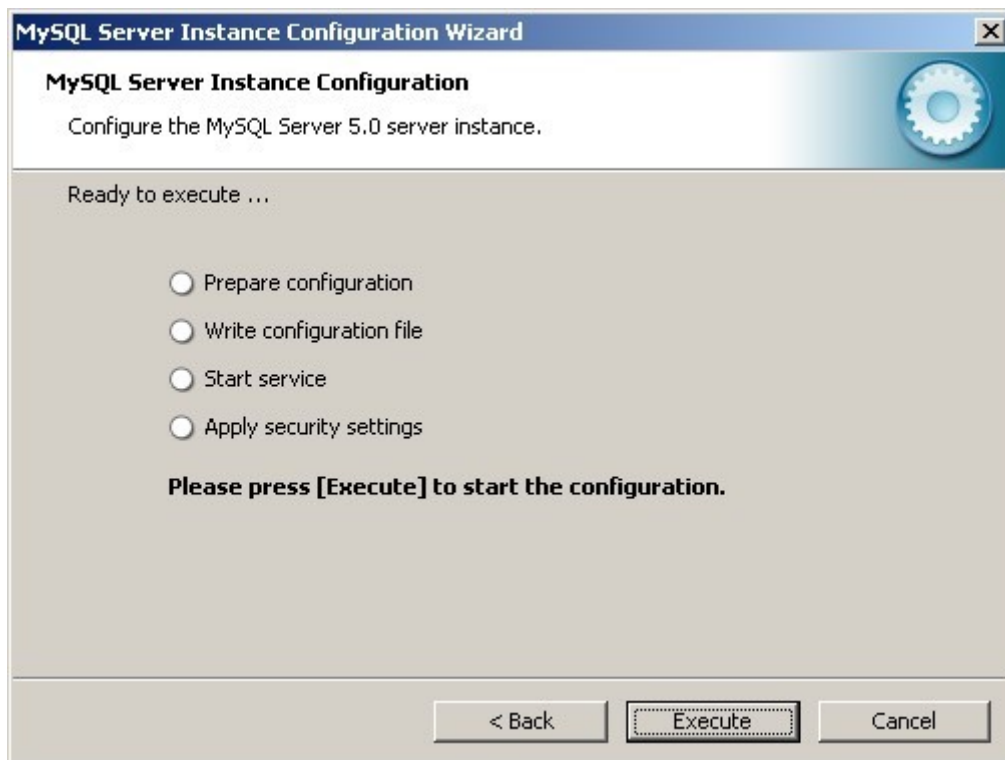
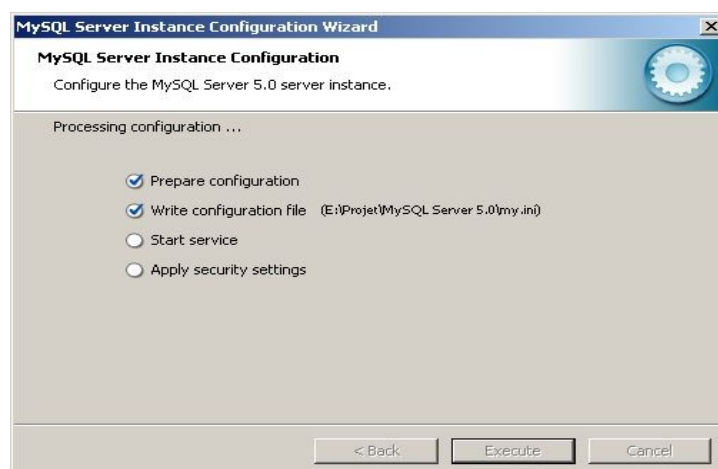


Illustration 41: Ecran configuration du serveur MySQL / On appuye sur Execute

On clique sur le bouton **Execute** pour lancer la configuration de ce dernier.



Lors de cette étape, il est possible que vous ayez un message d'erreur similaire à celui ci :

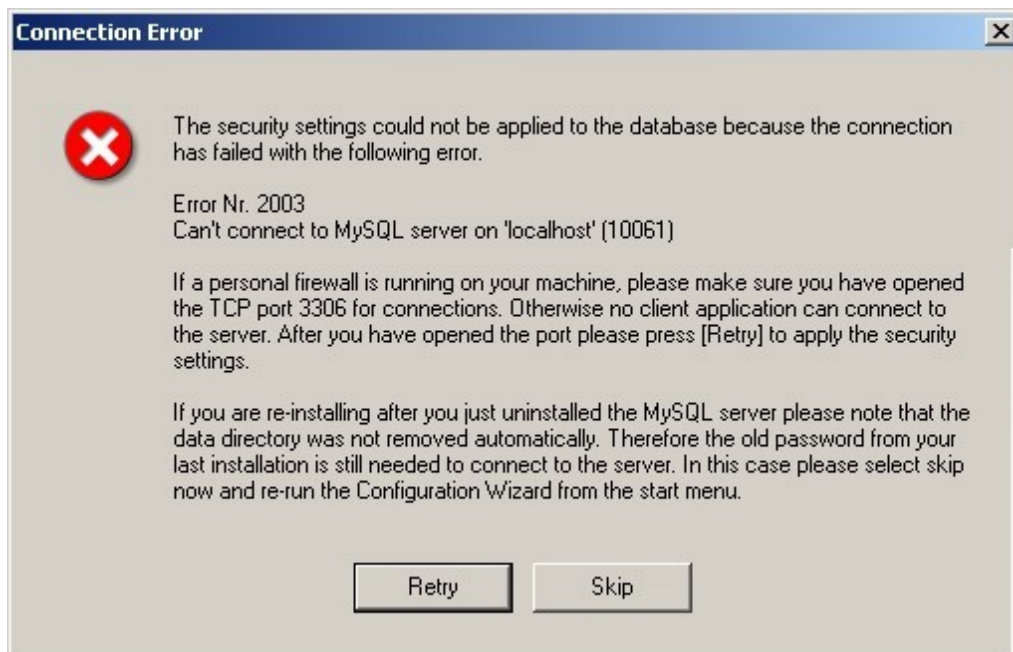


Illustration 42: Ecran message d'erreur problème de pare-feu

Dans ce cas là, il vous suffit d'ajouter une exception dans le pare-feu de Windows (si vous utilisez le pare-feu intégré à Windows XP SP 2). Pour configurer correctement le pare-feu, **Démarrer / Paramètres / Panneau de configuration / Centre de sécurité / cliquer sur Pare-feu Windows.**

De là, on clique sur l'onglet **Exceptions** dans la fenêtre du Pare-feu.

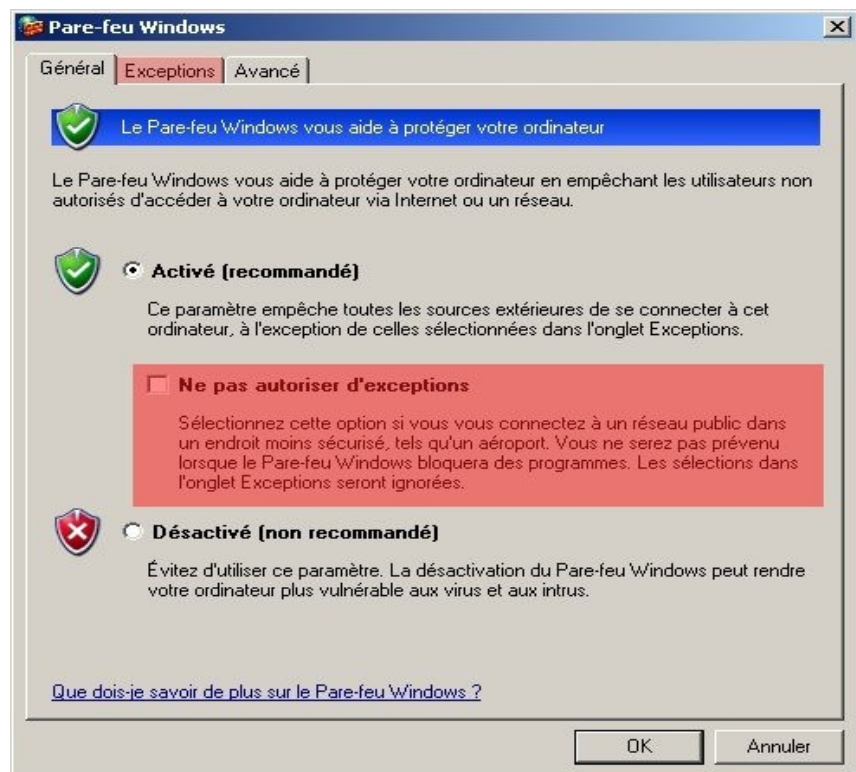


Illustration 43: Ecran pare-feu windows XP SP 2

Attention : la case « Ne pas autoriser d'exceptions » ne doit pas être cochée.

Dans la fenêtre **Exceptions**, cliquer sur le bouton **Ajouter un Port** et faite comme suit :

Nom : **MYSQL** ou **mysql**

Numéro de port : **3306** (port par défaut du serveur mysql)

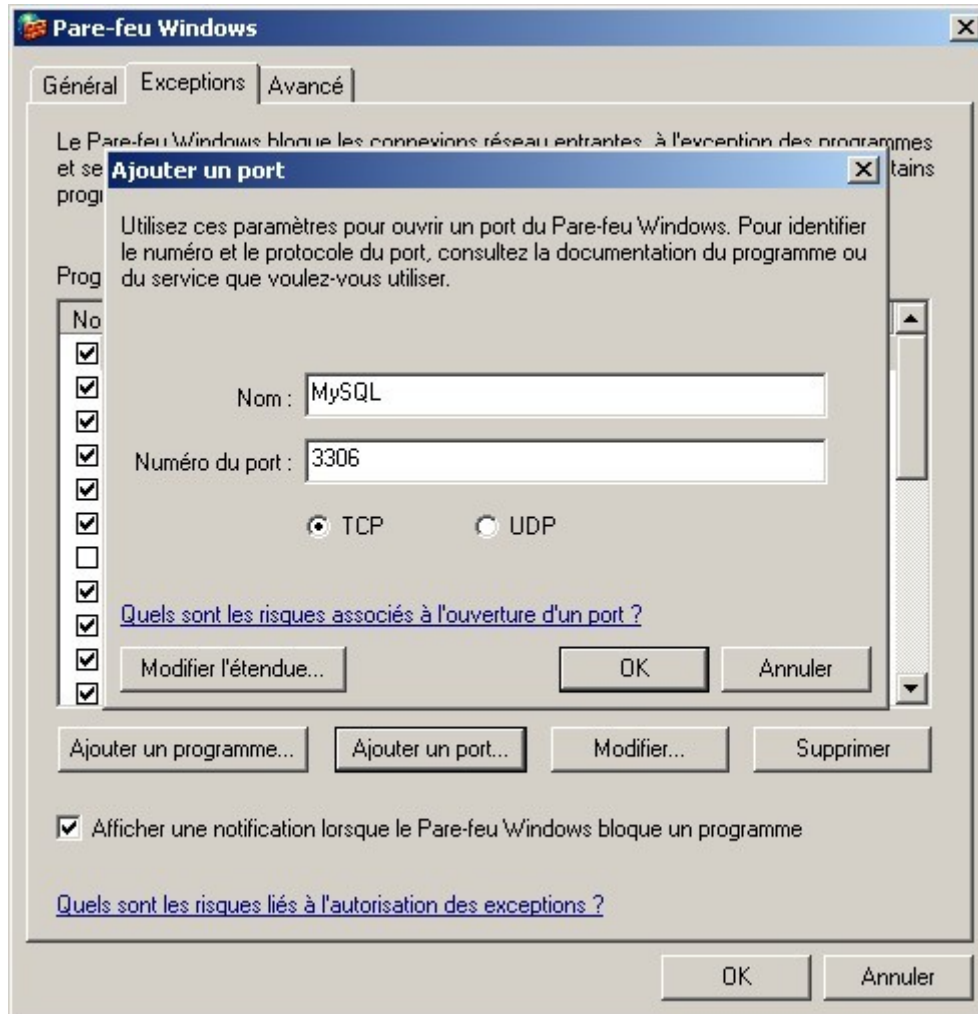


Illustration 44: Ecran onglet exceptions du pare-feu / Fenêtre Ajouter un port

Ceci fait, on clique sur le bouton **OK**, normalement, le nom a été ajouté à la liste des exceptions. On clique sur le bouton **OK** de la fenêtre **Exceptions** pour terminer l'opération.

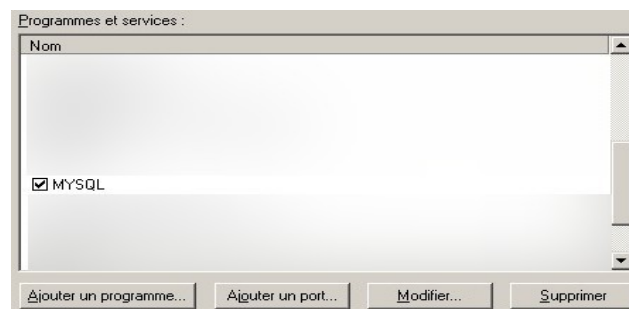


Illustration 45: Ecran liste des exceptions "programmes et services"

L'exception ajoutée, nous revenons à notre message d'erreur, nous cliquons sur le bouton **Retry**.



Illustration 46: Ecran message d'erreur pare-feu, on appuie sur le bouton Retry

Si la configuration du pare-feu est correcte, la configuration se poursuit sans problème.

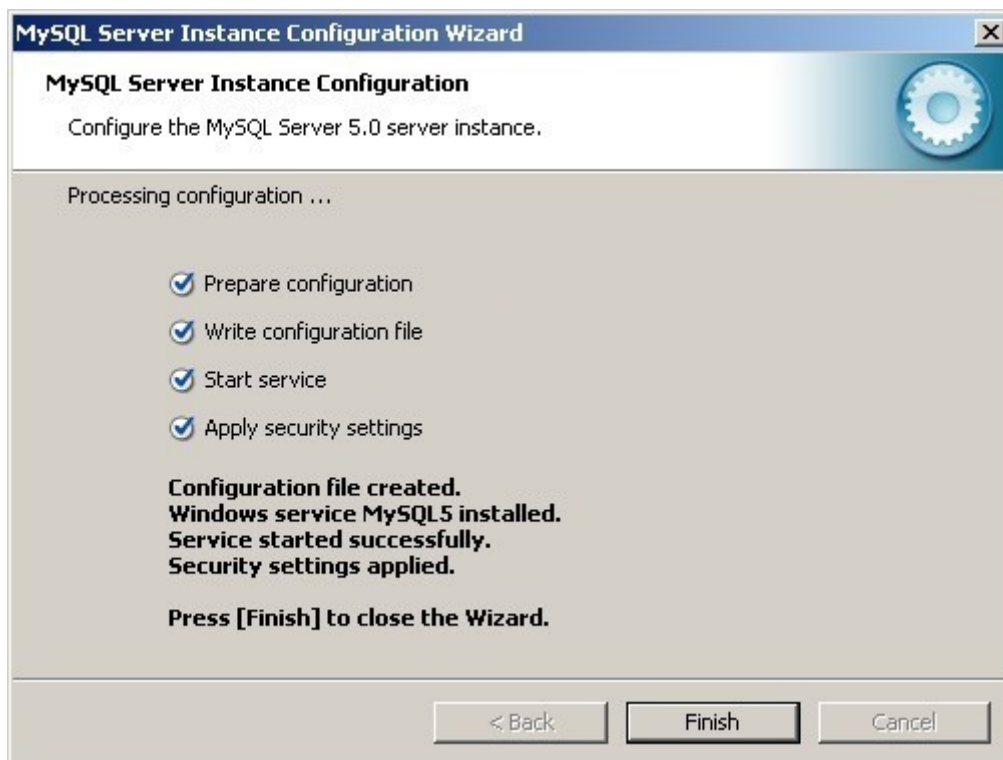


Illustration 47: Ecran fin de configuration de MySQL

L'installation et la configuration est à présent terminée, nous cliquons sur le bouton **Finish**.

Notre serveur MySQL ayant été installé en tant que service ce dernier doit être dès cet instant en action.

Par ailleurs, si vous regardez dans le menu Démarrer / Programmes, vous devriez avoir le menu suivant :

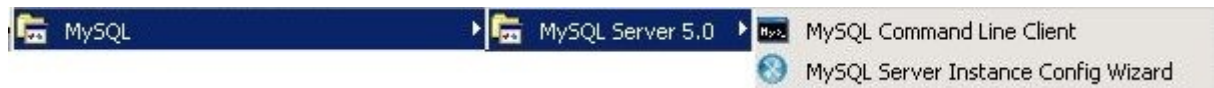


Illustration 48: Menu démarrer / Programmes / MySQL

Si nous souhaitons vérifier que notre serveur MySQL tourne bien, nous pouvons nous connecter au shell de ce dernier. Ainsi on clique sur **MySQL Command Line Client**.

Là, l'invite de commandes suivante s'affiche :

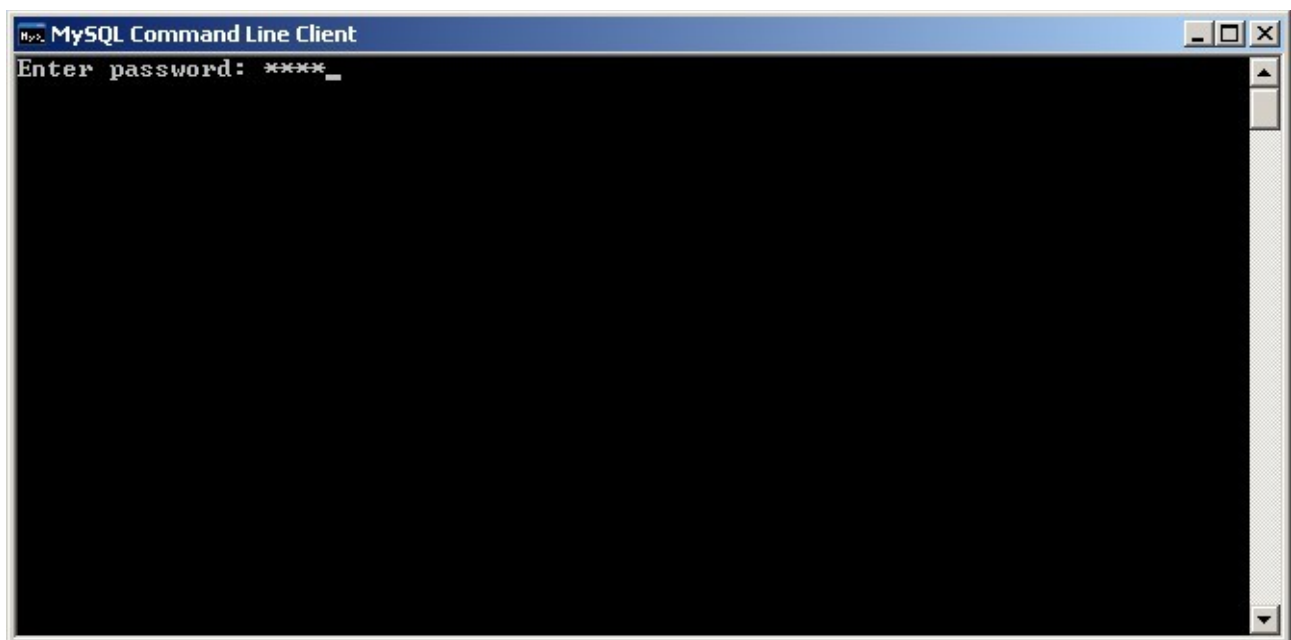


Illustration 49: Ecran MySQL Command Line Client demande de mot de passe root

On entre le mot de passe du compte **root** qu'on a défini lors de la configuration du serveur MySQL.

Nous voilà connecté sur le shell de notre serveur mysql.

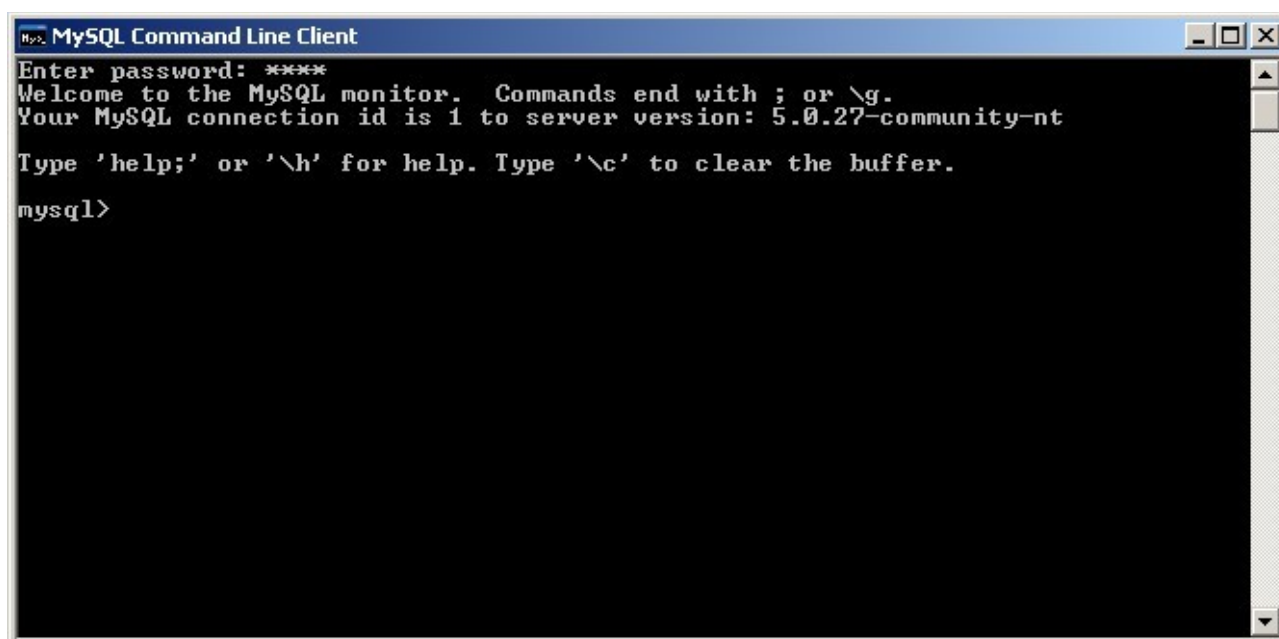


Illustration 50: Ecran MySQL Command Line Client connexion au serveur MySQL

Ce shell nous permet de surveiller notre serveur MySQL via quelques commandes, la liste des commandes est disponible en tapant **help**.

mysql> help

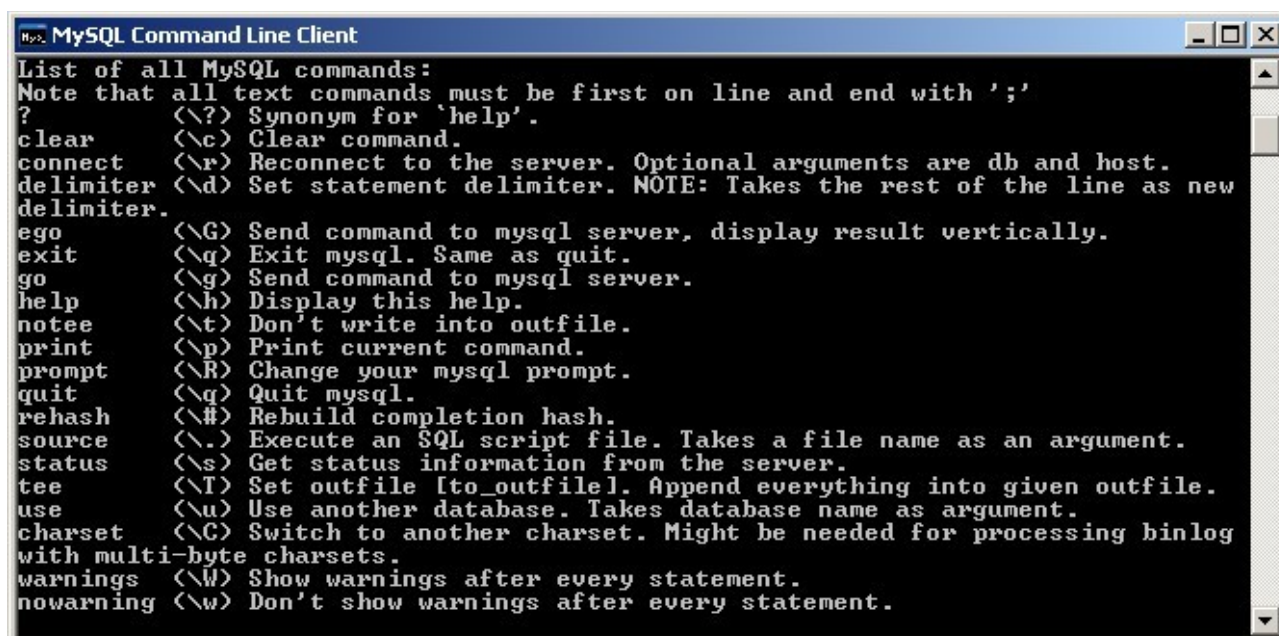
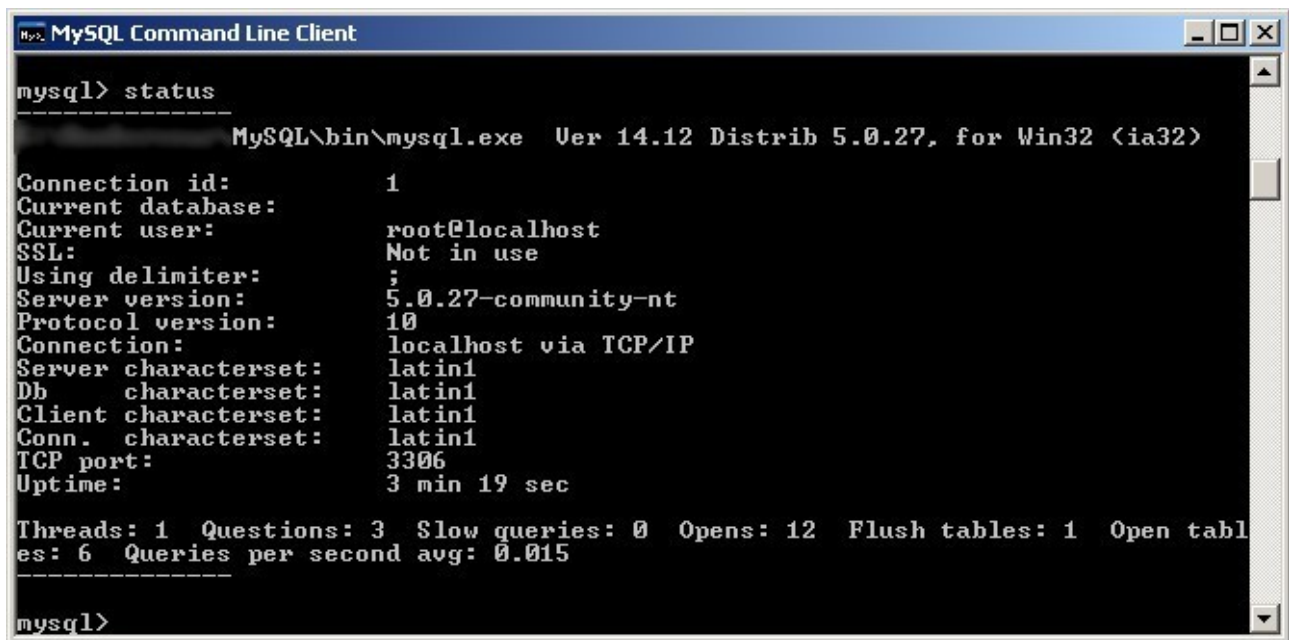


Illustration 51: Ecran MySQL Command Line Client résultat commande help - affichage des commandes de MySQL
L'écran d'aide qui s'affiche.

Exemple : **status**, cette commande va me fournir quelques informations concernant l'état de mon serveur mysql.



```
mysql> status
-----
MySQL\bin\mysql.exe  Ver 14.12 Distrib 5.0.27, for Win32 (ia32)

Connection id:          1
Current database:
Current user:           root@localhost
SSL:                    Not in use
Using delimiter:        ;
Server version:         5.0.27-community-nt
Protocol version:       10
Connection:             localhost via TCP/IP
Server characterset:    latin1
Db characterset:        latin1
Client characterset:    latin1
Conn. characterset:     latin1
TCP port:               3306
Uptime:                 3 min 19 sec

Threads: 1  Questions: 3  Slow queries: 0  Opens: 12  Flush tables: 1  Open tabl
es: 6  Queries per second avg: 0.015
-----
mysql>
```

Illustration 52: Ecran MySQL Command Line Client résultat commande status

En outre, j'obtiens l'uptime du serveur **3 min 19 sec**, le port utilisé **3306** (port par défaut de mysql).

Si nous souhaitons maintenant pouvoir administrer notre serveur mysql, nous pouvons le faire avec le programme **mysqladmin.exe** qui se situe dans le sous-répertoire **\bin** de notre installation.

Ce programme permet, en autres, la création/suppression de base/table, en bref toutes les commandes dont vous avez besoin pour l'administration de votre serveur.

Documentation officielle concernant **mysqladmin** disponible à l'url ci-dessous :

<http://dev.mysql.com/doc/refman/5.0/fr/mysqladmin.html>

Petit exemple de l'utilisation du programme **mysqladmin.exe**

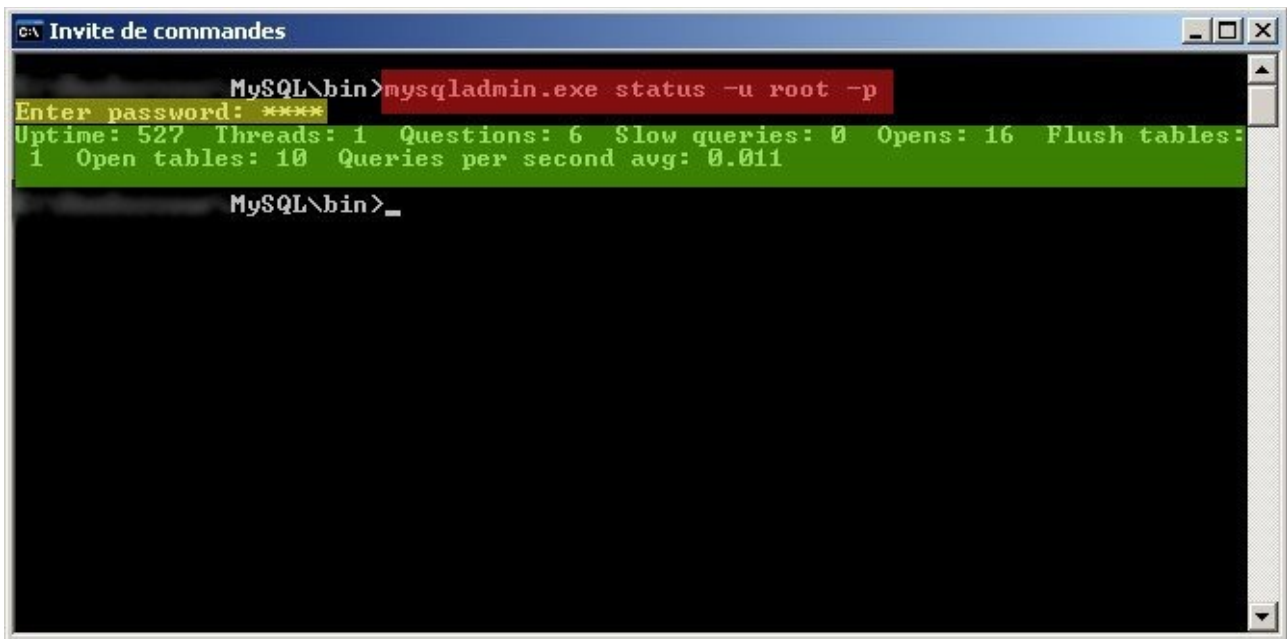


Illustration 53: Ecran invite de commandes résultat commande status par mysqladmin.exe

Dans le cas ci-dessous, je veux savoir l'uptime de mon serveur. Pour ce faire, j'utilise la commande suivante dans une invite de commandes :

mysqladmin.exe status -u root -p

-u root : permet de spécifier le nom d'utilisateur à utiliser pour se connecter si l'on ne souhaite pas utiliser celui par défaut (**ODBC**).

-p : permet d'avoir une demande de mot de passe pour le compte utilisateur rentré.

status : étant la commande que je souhaite exécuter.

Ceci n'est qu'un exemple simple de commande que l'on peut faire exécuter à notre serveur MySQL.

Un autre programme qui peut être intéressant de connaître c'est **mysqlshow**, cette commande permet d'afficher la liste des bases que contient notre serveur de données. Ce programme se trouve lui aussi dans le sous-répertoire **\bin** de notre installation.

Pour se faire j'utilise la commande suivante :

mysqlshow.exe -u root -p

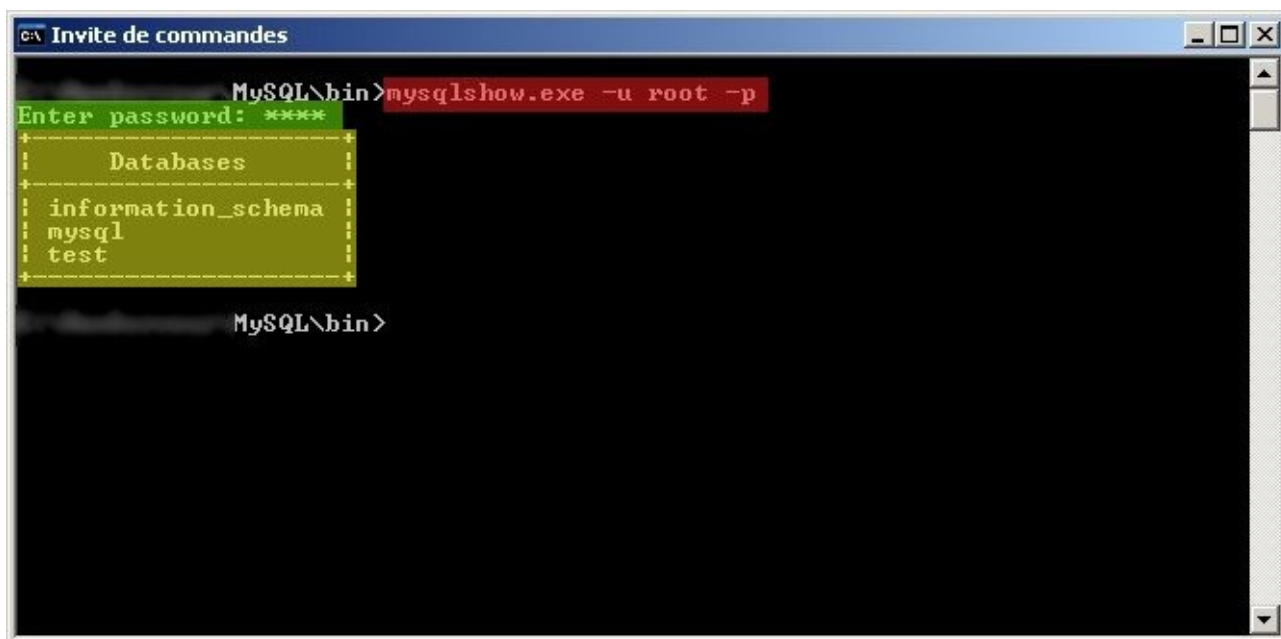


Illustration 54: Ecran invite de commandes résultat commande `mysqlshow.exe`

Dans le cas présent, nous avons donc **3** bases de données... L'une se nomme **information_schema**, l'autre **mysql** (base par défaut lors de l'installation de mysql) et la dernière **test**. C'est les trois bases de données par défaut lorsque l'on vient d'installer mysql. A noter que l'on peut très bien supprimer la base **test** qui ne servira pas dans bien des cas, pour réaliser la suppression, j'utilise la commande suivante :

`mysqladmin.exe drop test -u root -p`

Ceci étant, je vous encourage à lire la documentation officielle sur les différentes commandes que propose mysql.

Documentation officielle concernant **mysqlshow** disponible à l'url ci-dessous :

<http://dev.mysql.com/doc/refman/5.0/fr/mysqlshow.html>

Voilà en somme les informations qui vous permettront de réaliser les opérations les plus courantes sur votre serveur de données. Ci-dessous la documentation en français de la version 5.0 de mysql.

<http://dev.mysql.com/doc/refman/5.0/fr/index.html>

Ceci étant, un serveur mysql c'est bien mais l'administration de ce dernier via des lignes de commandes peut effrayer bon nombre de personnes.

Je vous propose donc de voir ci-dessous, quelques interfaces graphiques permettant l'administration de votre serveur simplement sans prise de tête.

Quelques interfaces graphiques :

Tout d'abord, nous commencerons avec le logiciel fourni par MySQL AB sur son site. Ils proposent un paquetage contenant différentes applications permettant l'administration d'un serveur mysql. Le paquetage en question se nomme **MySQL GUI Tools Bundle for 5.0**, il contient les logiciels suivants :

MySQL Administrator 1.2

MySQL Query Browser 1.2

MySQL Migration Toolkit 1.1

MySQL Workbench 1.1

Dans le cas présent, nous allons utiliser MySQL Administrator version 1.2.

Le paquetage est disponible à l'url suivante :

<http://dev.mysql.com/downloads/administrator/>

Au moment où je rédige ce document, nous sommes à la version **5.0r5** de **MySQL GUI Tools**. Il y a 2 archives de disponibles, soit la version avec installateur soit la version compressée sans installateur. Dans le cas présent, j'ai pris la version **compressée sans installateur** (taille **17,9 Mo** contre **17,6 Mo** pour la version avec installateur).



Illustration 55: Ecran de connexion de MySQL Administrator

Voici la fenêtre principale du programme MySQL Administrator.

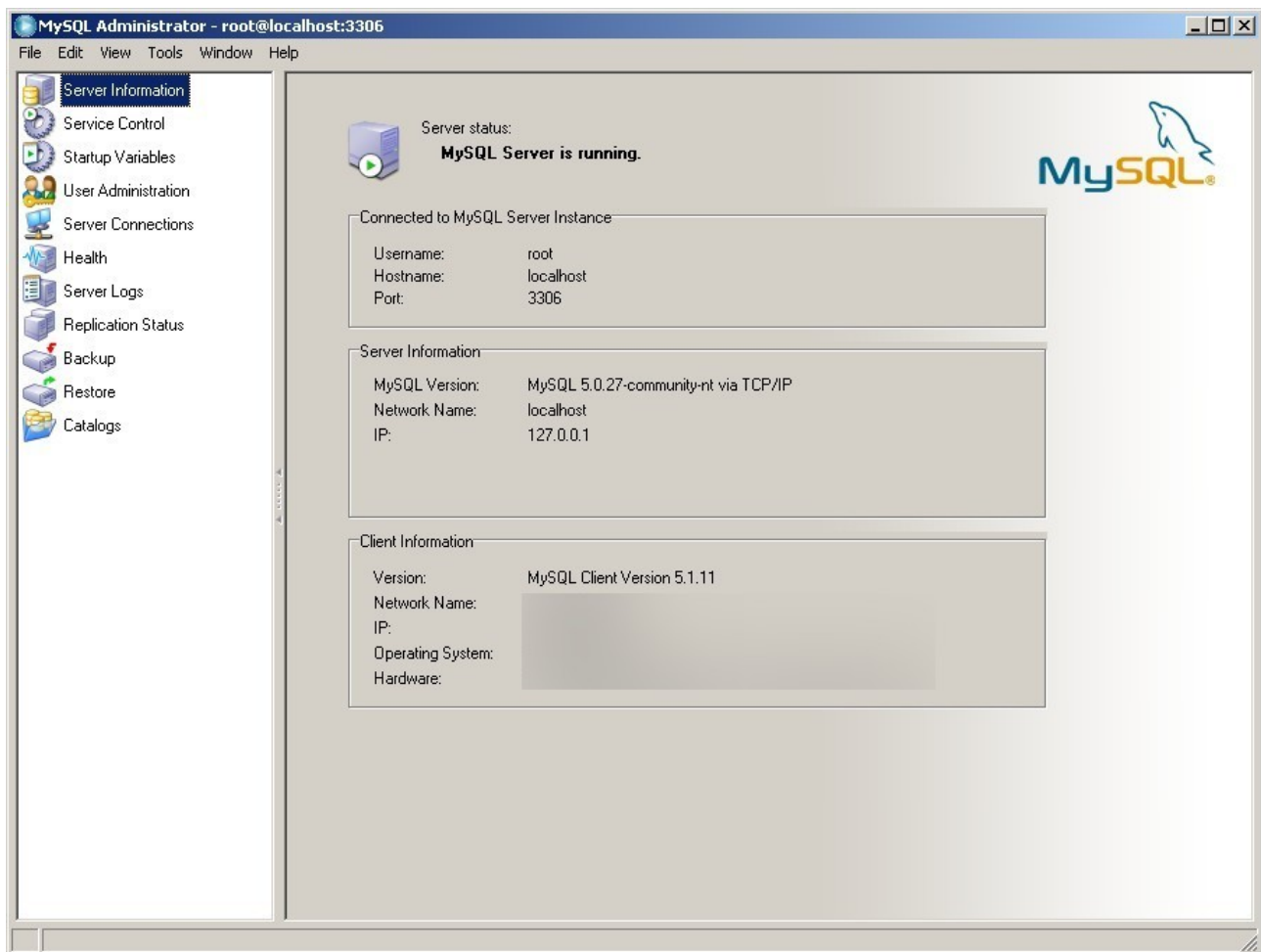


Illustration 56: Ecran principal de MySQL Administrator

MySQL Administrator vous permet d'administrer de façon simplifiée votre serveur de données. Entre autres, nous avons, la sauvegarde/restauration des données, l'ajout/suppression de compte utilisateur, la surveillance du serveur, l'affichage du journal, etc...

Par ailleurs, on peut réaliser la création / suppression de base, de table. Et ceci grâce à l'option **Catalogs**.

Une autre interface graphique disponible est **HeidiSQL**, ce logiciel est gratuit dans sa présente version (**3.0 RC3**).

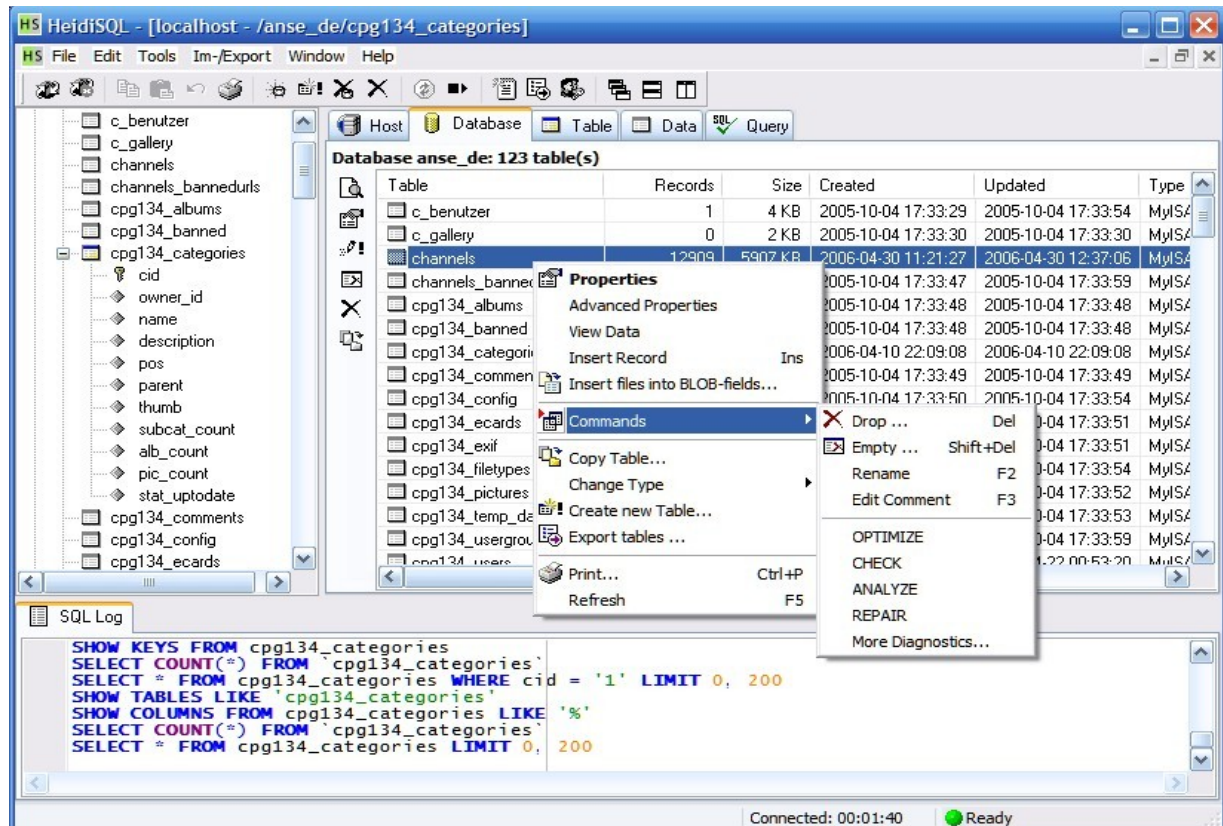


Illustration 57: Ecran principal de HeidiSQL – Origine de la capture : site officiel <http://www.heidisql.com>

Vous pouvez télécharger **heidiSQL** sur le site officiel.

Site officiel : <http://www.heidisql.com/> (in english)

NDLA : Personnellement, je n'ai pas tester ce programme donc je ne saurais donner un avis sur ce dernier.

Une dernière interface graphique que j'affectionne est [phpMyAdmin](#), qui est une interface web donc dans le cas présent, il faut que vous ayez un serveur web (par exemple, Apache) configuré pour PHP. Cette condition remplie, vous téléchargez la version courante, dans le cas présent, **phpMyAdmin 2.9.1**, celle-ci se présente sous la forme d'un fichier compressé (taille **3.5 mo**). Malheureusement, le jour où j'ai fini la mise à jour du tutoriel, il y avait un soucis pour télécharger sur le site officiel, donc, j'utiliserais une ancienne version pour exposer mes propos concernant l'installation et la configuration de phpmyadmin.

Une fois l'archive récupérée, on la décompresse pour le mieux dans un des sous-répertoires du serveur web. Et on renomme le répertoire en **phpmyadmin**.

Exemple : **e:\projet\phpmyadmin**

Ceci fait, nous devons éditer le fichier de configuration pour se faire, on fait une copie du fichier **config.default.php** que l'on nommera **config.inc.php**. On édite ce fichier avec le bloc-notes pour modifier quelques informations.

Voici les quelques éléments à modifier :

```
$cfg['PmaAbsoluteUri'] = '';
    en
$cfg['PmaAbsoluteUri'] = 'http://127.0.0.1/mysql/';

$cfg['blowfish_secret'] = '';
    en
$cfg['blowfish_secret'] = 'une phrase longue par exemple';

$cfg['Servers'][$i]['auth_type']    = 'config';
$cfg['Servers'][$i]['user']        = 'root';
    en
$cfg['Servers'][$i]['auth_type']    = 'cookie';
$cfg['Servers'][$i]['user']        = '';
```

Ceci fait on sauvegarde les modifications. Par ailleurs, dans votre fichier de configuration, il vous faudra ajouter les lignes suivantes (dans la partie Alias d' Apache) :

```
Alias /mysql/ "e:/projet/phpmyadmin/"
```

```
<Directory "e:/projet/phpmyadmin">
```

```
Options Indexes MultiViews
```

```
AllowOverride None
```

```
Order allow,deny
```

```
Allow from all
```

```
</Directory>
```

On sauvegarde le fichier de configuration « **httpd.conf** » après avoir ajouter les quelques lignes (ci-dessus). Une fois ceci fait, il vous faudra redémarrer votre serveur Apache.

Dans votre navigateur favori (Firefox par exemple) on tape dans la barre d'adresse :

<http://127.0.0.1/mysql/> ou <http://localhost/mysql/>

Vous devriez avoir un écran similaire à ceci :



Illustration 58: Ecran de Firefox qui affiche la page d'authentification de phpmyadmin

Entrez votre nom d'utilisateur (*root* par exemple) et votre mot de passe...

Ci-dessous, l'interface d'administration du serveur de données.

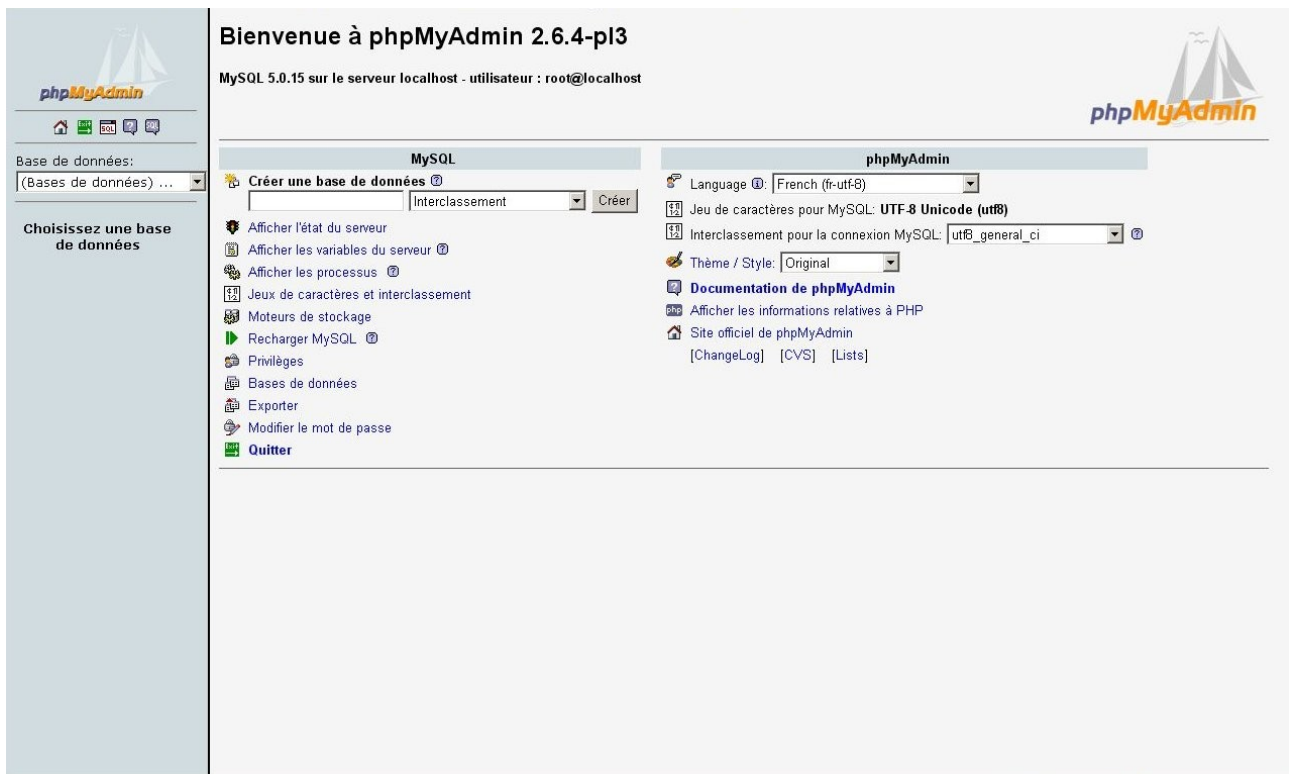


Illustration 59: Ecran de Firefox qui affiche la page principale de phpmyadmin

Cette interface web, vous permet de réaliser toutes les opérations d'administration/maintenance sur votre serveur de données, entre autres, création/suppression d'utilisateur, création/suppression de bases, de tables. De quoi ravir toutes les personnes souhaitant administrer au mieux leur serveur mysql.

Site officiel : <http://www.phpmyadmin.net> (in english)

Voilà, à présent, vous avez quelques informations pour faire votre choix en matière d'interface d'administration pour votre serveur de données.

Rendre accessible son serveur web :

Votre serveur Apache / MySQL installé et configuré, il ne vous reste plus qu'à le rendre accessible au plus grand nombre. Vous avez plusieurs possibilités.

La première option que vous avez c'est d'utiliser l'adresse IP de votre connexion internet, cette option est à réserver si l'on dispose d'un IP dite fixe (à savoir qu'elle ne change jamais). L'IP fixe est disponible gratuitement chez certain F.A.I (Fournisseur d'Accès à Internet). Pour connaître l'IP de votre connexion, vous pouvez faire un tour à l'url suivante :

<http://ip.monserveurperso.com/>

La deuxième option qui est disponible, c'est l'utilisation d'un sous-domaine disponible gratuitement chez dyndns.org ou no-ip.com (en autres). Ce sous-domaine sera de la forme suivante : **vous.no-ip.com** ou **vous.dyndns.org**. La création d'un compte chez l'un ou chez l'autre est gratuite. Pour que votre sous-domaine corresponde à votre IP internet du moment, vous devrez **installer un client** sur votre ordinateur. Ce dernier **mettra à jour automatiquement** votre IP sur le site où vous avez votre sous-domaine. Ceci afin de permettre **l'accessibilité de votre serveur** simplement et facilement. Pour plus d'informations, vous pouvez consulter le tutoriel à ce sujet à l'url suivante :

<http://tutorial.monserveurperso.com/Dns/>

La troisième et dernière option est l'achat d'un nom de domaine. Un nom de domaine selon le registrar vous coûtera dans les 14 euros ttc / an (chez Gandi). Ce dernier pourra être de la forme suivante : **www.vous.com**. Une fois l'acquisition du domaine effectuée, on définira les serveurs DNS chez le registrar. Certains registrars permettent l'utilisation du nom de domaine sur IP dynamique, dans ce cas là, il propose un client qui mettra à jour votre IP sur le site de ce dernier. L'autre solution étant d'être son propre serveur DNS ou de faire appel à un intervenant tiers qui remplira cette fonction. Le serveur DNS étant celui qui dit « telle adresse internet, exemple **www.vous.com** » = « telle adresse ip ». Certains services tels que zoneedit.com propose de prendre en charge la partie serveur DNS, ce qui vous laisse champ libre pour configurer à votre goût votre domaine. L'autre solution étant d'héberger vous-même votre serveur DNS primaire le secondaire pouvant être votre registrar. Si vous souhaitez paramétrer un nom de domaine sur une ip dynamique, vous pouvez consulter le tutoriel consacré à ce sujet à l'url suivante :

[Installation et configuration d'un nom de domaine pour une IP dynamique](#)

Dans tous les cas, il vous faudra configurer votre modem/routeur ou votre box (livebox, freebox, 9box, alicebox, etc..) **pour que le port 80 soit redirectionné** vers la machine qui fait office de serveur. Par ailleurs penser à configurer votre pare-feu (Windows ou autres) pour que le port 80 soit accepté.

Ressources :

Quelques url's :

Apache : <http://www.Apache.org>

PHP : <http://www.php.net>

MySQL : <http://www.mysql.com>

MySQL en français : <http://www-fr.mysql.com/>

PhpMyAdmin : <http://www.phpmyadmin.net>

MySQL GUI Tools : <http://dev.mysql.com/downloads/administrator/>

HeidiSQL : <http://www.heidisql.com/>

Définition de MySQL sur Wikipédia : <http://fr.wikipedia.org/wiki/Accueil>

Définition de PHP sur Wikipédia :

http://fr.wikipedia.org/wiki/PHP_hypertext_preprocessor

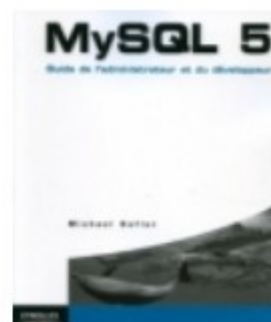
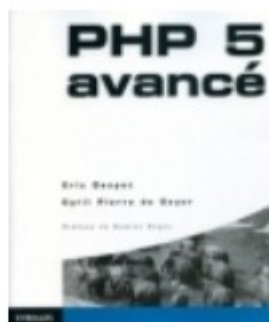
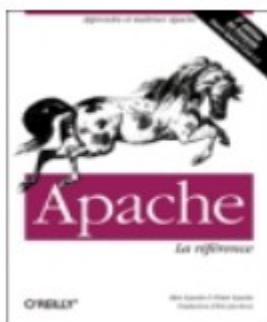
Définition d'Apache sur Wikipédia : http://fr.wikipedia.org/wiki/Apache_HTTP_Server

Documentation MySQL en français : <http://dev.mysql.com/doc/refman/5.0/fr/index.html>

Documentation Apache en français : <http://httpd.apache.org/docs/2.0/>

Documentation PHP en français : <http://fr.php.net/manual/fr/>

Quelques livres :



Conclusion :

Vous avez à présent un serveur web opérationnel que vous allez rendre disponible sur internet et ce de manière simple. Il est à noter que la configuration du serveur quoiqu'un peu ardue au début devient vite une chose acquise pour peu qu'on se donne la peine de lire le manuel.

Si vous avez des questions / suggestions concernant ce document, je vous encourage à me contacter soit par e-mail à thecyberseb@hotmail.com, soit en laissant un message sur le forum qui se trouve à l'url suivante :

<http://forum.monserveurperso.com>

Pour de plus amples documents sur divers sujets, faites un tour à l'url ci-dessous...

<http://tutorial.monserveurperso.com>

Merci !

A **Daria** (pour la relecture ainsi que pour la correction orthographique).

A **Pat** (pour la relecture ainsi que pour la correction orthographique).

A **Salvador** (pour la relecture ainsi que pour la correction orthographique).

A **Framasoft.net** qui a diffusé la première version de ce document.

A **Skweek** (pour son amitié de tous les jours)

Aux **sites internet** qui publient avec ou sans ma bénédiction mon document.

Et **merci à vous, anciens et nouveaux lecteurs**
qui avait fait de ce document une référence de nos jours.

MERCI d'avoir pris le temps de lire ce document :-)