

Serveur WIMP sous Windows 7 dans le cadre de la mise en place d'un serveur local de développement

par POINSOT Benjamin (Benj.)

Date de publication : 31 juillet 2011

Dernière mise à jour : 03 août 2011

Lorsque l'on pense au développement Web, on y associe bien souvent l'acronyme LAMP (Linux, Apache, MySQL et PHP). Seulement, depuis Windows Vista, IIS est devenu un très bon serveur Web de par le nombre croissant de fonctionnalités ajoutées. Il est intéressant de pouvoir l'utiliser lorsqu'on ne souhaite pas s'orienter vers un système d'exploitation autre.

Ce tutoriel va, vous l'aurez compris, vous donner la démarche à suivre afin de mettre en place un serveur local de développement basé sur IIS (dans sa version 7.5 sous Windows 7), MySQL v5.5.14 et PHP v5.3.6 (dernières versions en date au moment de la rédaction de ce tutoriel).

I - Introduction.....	3
II - Installation de IIS.....	4
III - Téléchargement et mise en place de PHP.....	6
IV - Installation et configuration de MySQL.....	17
V - Installation et configuration de PHPMyAdmin.....	30
VI - Filtrage des accès à votre site Web.....	33
VII - Mise en place d'un certificat SSL auto-généré pour votre site Web.....	35
VIII - Forcer l'utilisation du SSL avec PHPMyAdmin.....	39
IX - IIS et la réécriture d'adresses (aka URL rewriting).....	40
X - Lire les journaux de votre site Web.....	43
XI - Conclusion.....	47

I - Introduction

Un serveur WIMP est un serveur Web qui utilise Windows comme système d'exploitation, IIS comme serveur Web, MySQL comme serveur de base de données et PHP comme langage interprété.

Windows 7 (dans son édition professionnelle) est utilisé ici mais vous retrouverez exactement les même écrans (hormis l'**Ajout de fonctionnalités Windows** sous Windows 7 qui sera intitulé **Ajout de rôles** sous Windows 2008 Server).

Dans la plupart des cas, les serveurs qui utilisent PHP sont des serveurs sous Linux mais il n'est pas toujours très aisé d'utiliser Linux lorsque les bases peuvent faire défaut. IIS, depuis la version 6 (sous Windows 2003 Server), a rendu l'utilisation de PHP très facile et c'est d'autant plus vrai maintenant qu'il gère désormais (depuis la version 7) la réécriture d'adresses (URL rewriting) ainsi que l'importation de règles depuis les fichiers .htaccess. Vous trouverez également, sur le site de Microsoft dédié à IIS, une pléthore de composants additionnels vous permettant de mettre en place différentes choses intéressantes si vous faites du développement.

Nous allons nous limiter ici à un serveur de test local avec différentes options basiques. Il ne s'agit en aucun cas de mettre ceci en place pour un serveur de production. En effet, nous n'aborderons aucunement la sécurité (ou très légèrement) des différents serveurs et des différentes configurations. Par exemple, pour PHP, il convient de désactiver **allow_url_fopen** qui est actif par défaut ou bien paramétrer le pare-feu de Windows, etc.

Ce tutoriel, vous l'aurez compris, part du principe que vous êtes en possession d'un système d'exploitation de type Windows (7 ou 2008 Server donc) pleinement fonctionnel. Le cas échéant, je vous invite à consulter un (parmi d'autres) article de PC Astuces à cette adresse : http://www.pcastuces.com/pratique/windows/7/installer_windows7/page1.htm afin d'installer le système d'exploitation qui fait l'objet de ce tutoriel.

Voici comment nous allons procéder pour mettre en place ce serveur de test : dans un premier temps, nous allons installer Internet Information Services (que vous retrouverez dans ce tutoriel sous la forme abrégée "IIS"). Ensuite, nous nous occuperons de télécharger, "d'installer" et de configurer PHP puis nous intégrerons PHP à IIS. Suivra l'installation de MySQL puis nous terminerons par quelques petites astuces et conseils pratiques.


II - Installation de IIS

Afin d'installer IIS, il vous faut aller dans le **Panneau de configuration**. Contrairement aux précédentes versions d'IIS (je pense plus particulièrement aux version 5 et 6), vous n'avez pas besoin du CD/DVD d'installation de Windows pour installer IIS.

Cliquez maintenant sur **Programmes**.

A présent, cliquez sur **Activer ou désactiver des fonctionnalités Windows**.

Dans la liste des fonctionnalités proposées, cocher la case correspondant à **Service Internet (IIS)** et uniquement cette case (configuration par défaut pour un serveur IIS, nous installerons plus tard les composants qu'il nous manque).

 Services Internet (IIS)

L'installation une fois lancée, patientez quelques instants.

Une fois l'installation terminée, vérifiez que le serveur IIS est bien installé et fonctionne correctement en vous rendant à l'adresse suivante via votre navigateur : <http://localhost>.

Une page telle que celle-ci-dessous devrait être affichée :

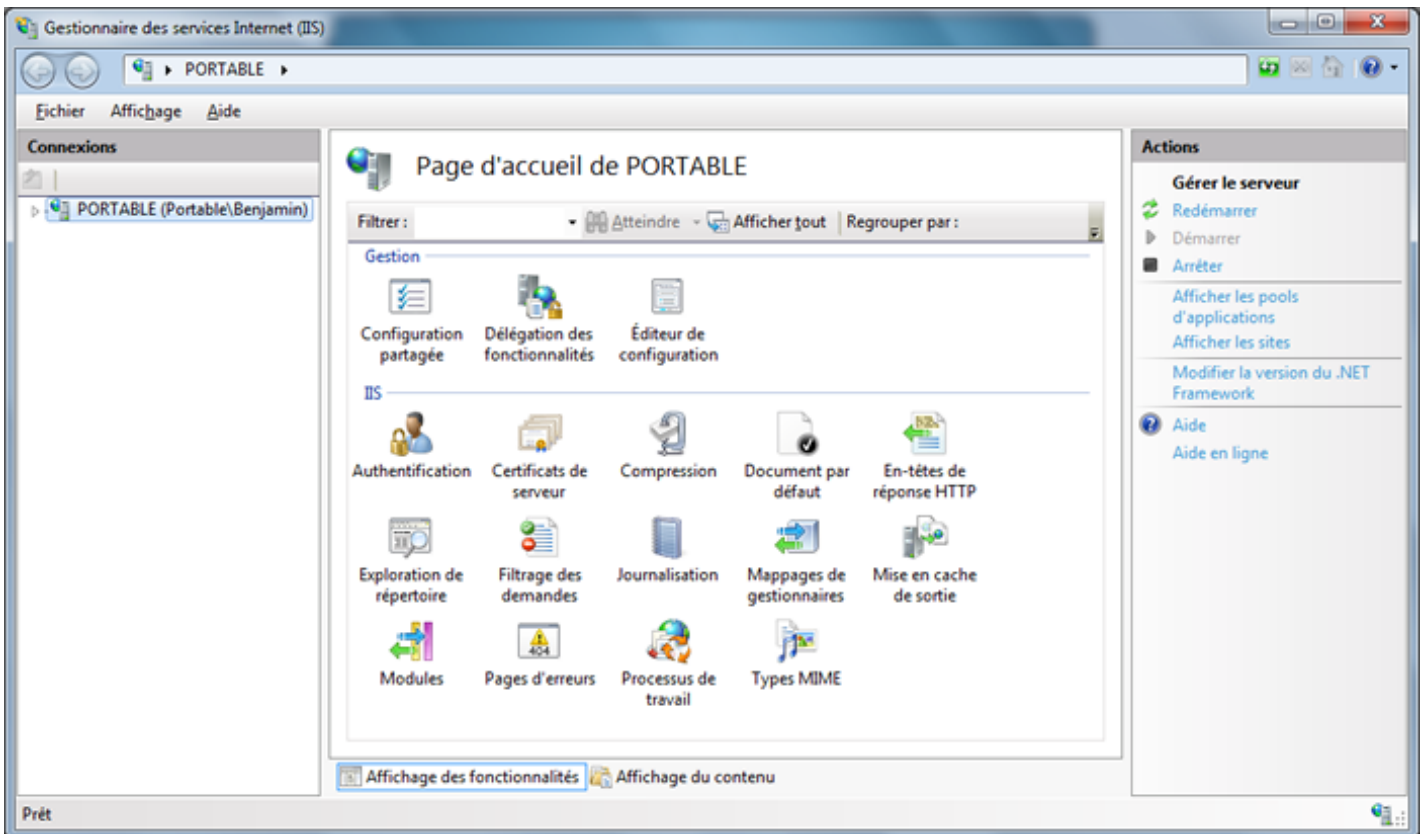


Ceci étant fait, nous allons maintenant nous rendre dans la console de gestion d'IIS. Pour ce faire, retournez dans le panneau de configuration puis cliquez sur **Système et sécurité**.

Sélectionnez ensuite **Outils d'administration**.

Maintenant, choisissez **Gestionnaire des services Internet (IIS)**.

La console doit maintenant être ouverte, une fenêtre similaire à celle-ci-dessous :



Ne fermez pas cette fenêtre, nous allons bientôt en avoir besoin.

La première étape de ce tutoriel est terminée. Comme vous venez de le voir, l'installation du serveur Web IIS n'est qu'une simple formalité.

Passons à l'étape suivante si vous le voulez bien ; celle-ci concerne PHP.

III - Téléchargement et mise en place de PHP

Nous allons maintenant télécharger PHP dans sa dernière version stable, à savoir PHP v5.3.6.

Allez sur le site de PHP (à savoir <http://php.net/>) puis, en haut à droite, cliquez sur **Current PHP 5.3 Stable** :

Dans la page nouvellement affichée, au centre, cliquez sur le lien <http://windows.php.net/download/> de la section **Windows Binaries**. Si je vous fait passer par l'index du site de **PHP.net** et non par l'adresse directe pour les binaires Windows, c'est uniquement dans le but de s'assurer que la dernière version stable pour Windows et celle dont les sources proposées sur l'index sont les mêmes. En effet, comme vous pouvez le constater, il y a un délai de quelques jours entre les deux. Autant profiter de la dernière version disponible.

Dans la liste des packages disponibles, nous allons opter pour la suivante :

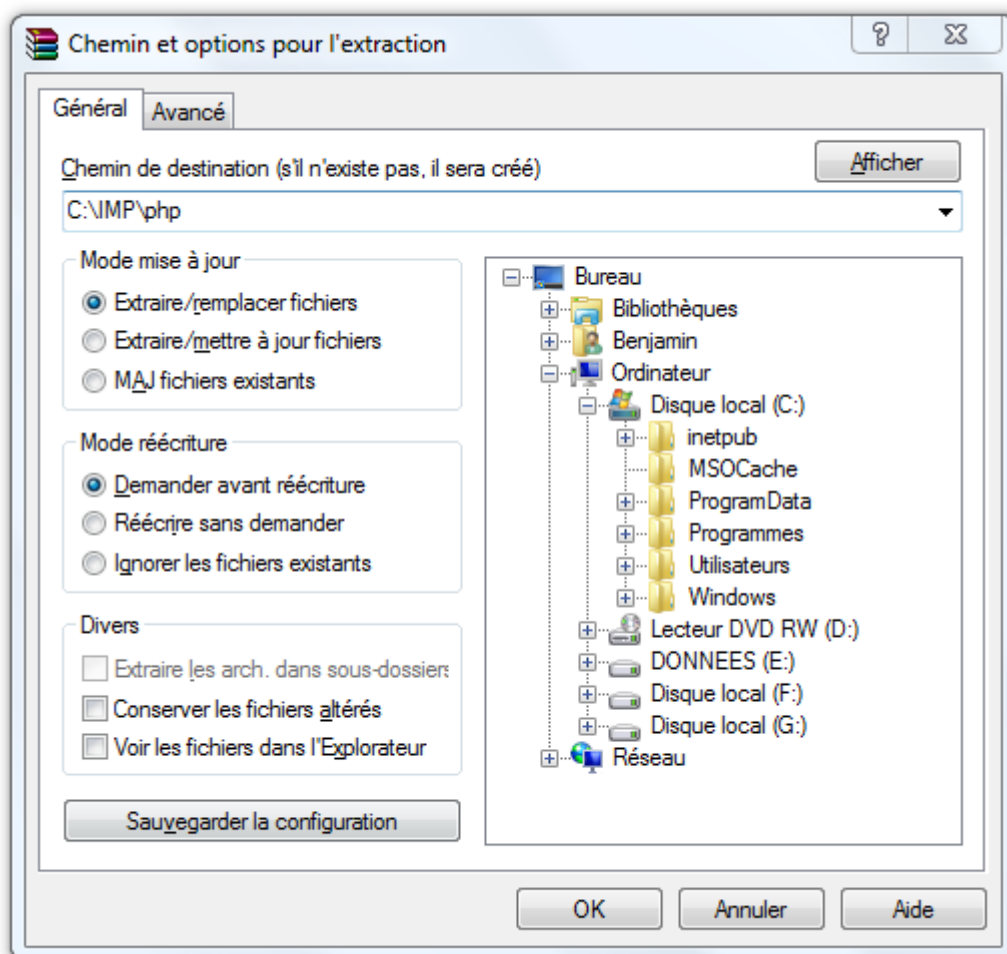
```
VC9 x86 Thread Safe (2011-Mar-22 13:27:32)  
  
▪ Zip [15.13MB]  
  
sha1: 3b385097d8ef61b01e0511e75c0c96353c079729
```

Version **VC9** en **Thread Safe** et surtout version **ZIP**. Je préfère en effet cette version, aucune installation n'est à faire, pas de clef en plus dans le registre, une simple extraction de fichiers et après on fait tout à la main (et ceux qui ne disposent pas d'une connexion très haut débit - il y en a encore, vous pouvez me croire - en seront d'autant plus content, l'archive est ce qu'il y a de plus léger à télécharger concernant PHP). D'autant plus qu'une version de type MSI vous oblige quand même à éditer le fichier de configuration de PHP (au moins dans le cadre de l'utilisation de PHP avec CGI).

Maintenant que l'on a sous la main PHP, il va falloir le faire fonctionner avec IIS (nous avons déjà le W de Windows, le I de IIS, restent le M de MySQL et le P de PHP).

Poursuivons donc avec PHP.

Le téléchargement effectué, nous allons extraire le contenu de l'archive à la racine du disque système dans un dossier que je vais nommer **IMP**. Vous pouvez adapter, bien sûr. Ce dernier sera le dossier contenant, vous l'aurez compris, MySQL et PHP (j'aime avoir les deux sous la main, au même endroit, libre à vous d'adapter) :



Je vais également créer dans ce dossier **IMP** un répertoire **www** dans lequel seront stockées toutes mes pages Web. Comme dit plus haut, c'est plus pratique de tout avoir à portée de main.

Si vous souhaitez modifier quelque paramètre que ce soit, libre à vous de le faire (ça ne pose aucun problème) mais pensez bien à adapter les différents chemins mentionnés tout au long de cet article.

Le contenu de l'archive étant extrait, nous allons passer à la configuration de PHP. Tout d'abord, renommez le fichier **php.ini-production** (**C:\IMP\php\php.ini-production** en ce qui me concerne) en **php.ini**.

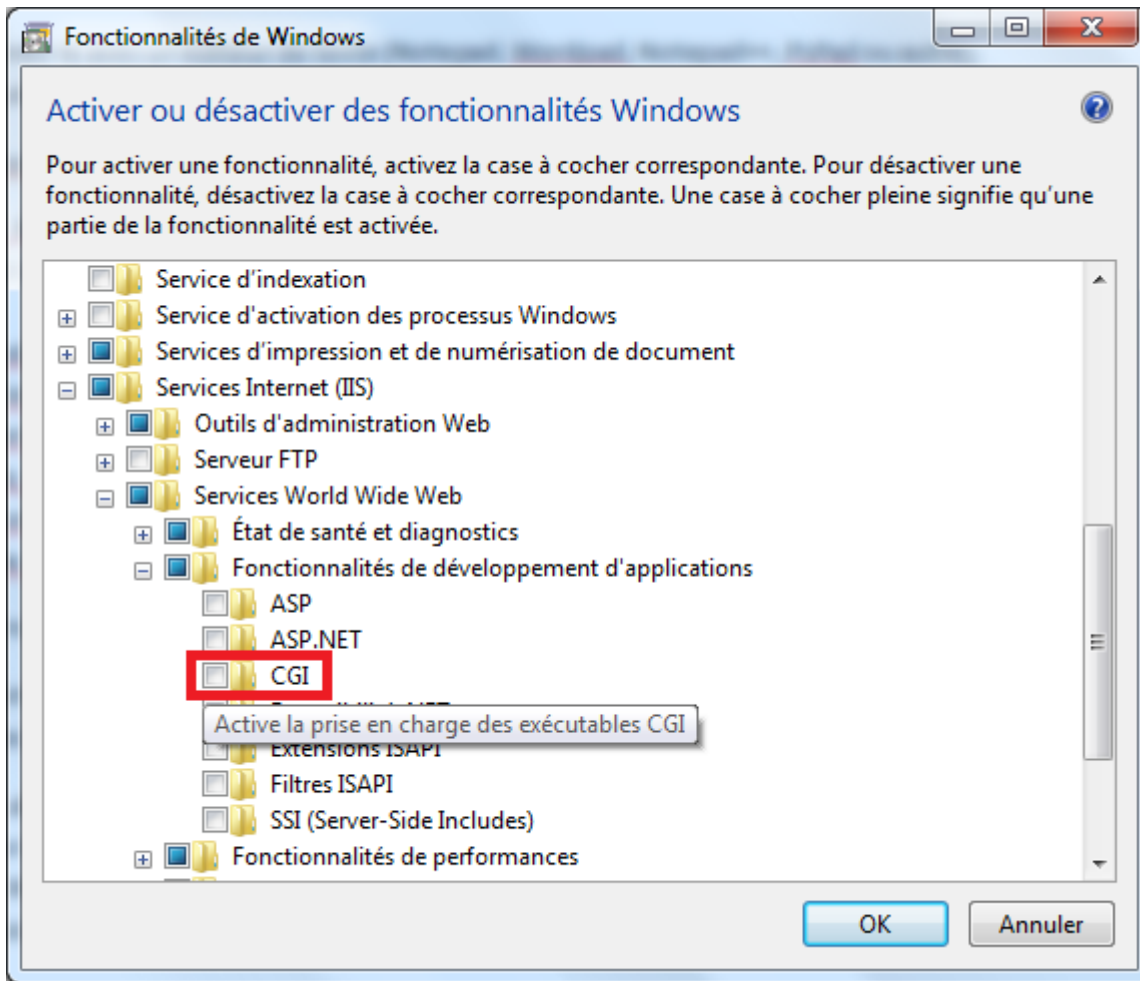
Maintenant ouvrez-le avec un éditeur de texte (Notepad, Wordpad, Notepad++, PsPad ou autre). Nous allons y opérer quelques changements :

- modifier «;open_basedir =» par «http://windows.php.net» (répertoire de "base" de votre serveur)
- «;extension_dir = "ext"» doit être changé par «extension_dir = "C:\IMP\phplext"» (il s'agit ici d'indiquer le répertoire contenant les extensions nécessaires à PHP).
- «;cgi.force_redirect = 1» doit également être modifié par «cgi.force_redirect = 0»
- «;cgi.fix_pathinfo=1» doit être adapté pour devenir «cgi.fix_pathinfo = 1»
- «;fastcgi.impersonate = 1;» doit être changé également par ceci : «fastcgi.impersonate = 1» (question de sécurité)
- «;upload_tmp_dir =» va quant à lui être remplacé par «upload_tmp_dir = "C:\Windows\Temp"» (répertoire dans lequel seront stockés les fichiers temporaires créés lors d'un envoi de fichier)
- «;extension=php_mbstring.dll» doit devenir «extension=php_mbstring.dll» (nécessaire, par exemple, à PHPMyAdmin)
- «;extension=php_gd2.dll» devient «extension=php_gd2.dll» (permet de générer des images)
- «;extension=php_mysql.dll» devient «extension=php_mysql.dll» (primordial pour une utilisation de MySQL via PHP)
- «;extension=php_mysqli.dll» devient «extension=php_mysqli.dll» (idem)
- «;date.timezone =» va être remplacé par «date.timezone = Europe/Paris» (vous l'aurez compris, il s'agit ici du fuseau horaire)
- «;session.save_path = "/tmp"» va devenir «session.save_path = "C:\Windows\Temp"» (répertoire dans lequel seront stockés les fichiers de session)

Ces modifications apportées, enregistrez le fichier et fermez votre éditeur de texte.

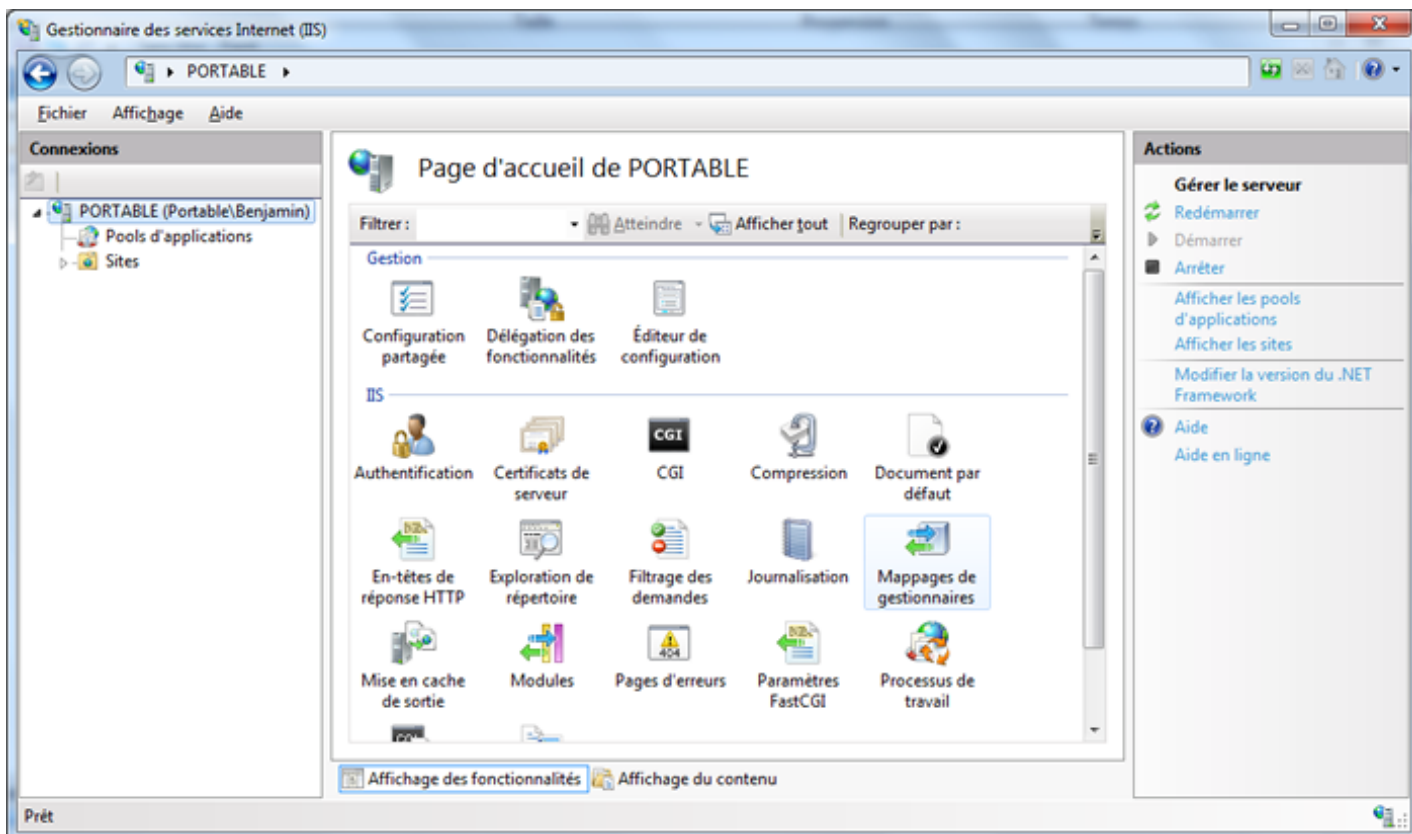
Nous allons maintenant passer à l'étape la plus importante de ce tutoriel : permettre à IIS de "travailler" avec PHP. Pour cela, nous allons installer un module pour IIS qui nous permettra d'utiliser PHP en FastCGI.

Allez dans le **Panneau de configuration**, dans **Programmes** et choisissez **Activer ou désactiver des fonctionnalités Windows**. Dans la fenêtre qui s'offre à vous, naviguez jusqu'à trouver **Services Internet (IIS)** puis cliquez sur le petit +.

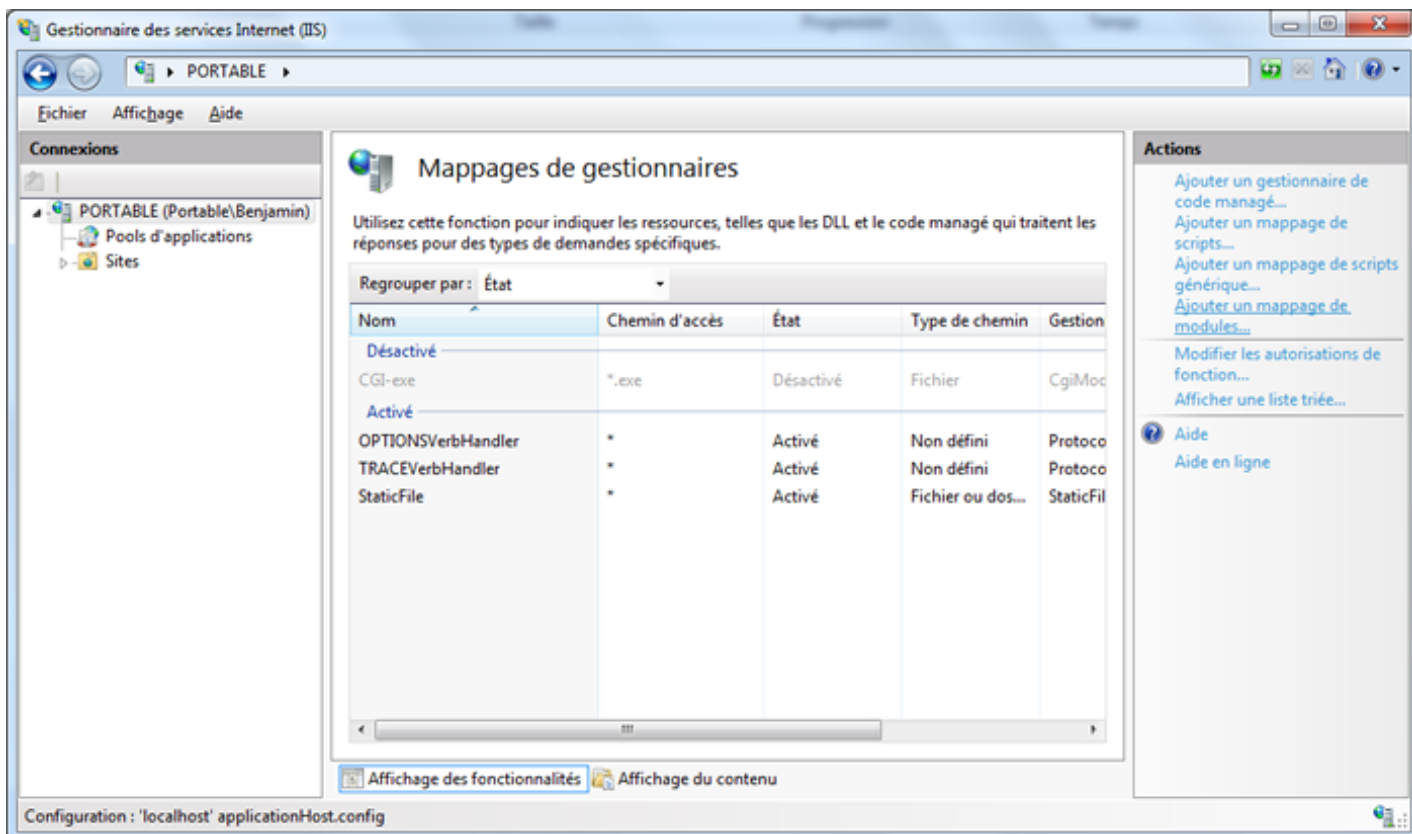


Cliquez dans la case **CGI** puis validez en cliquant sur le bouton **OK**.

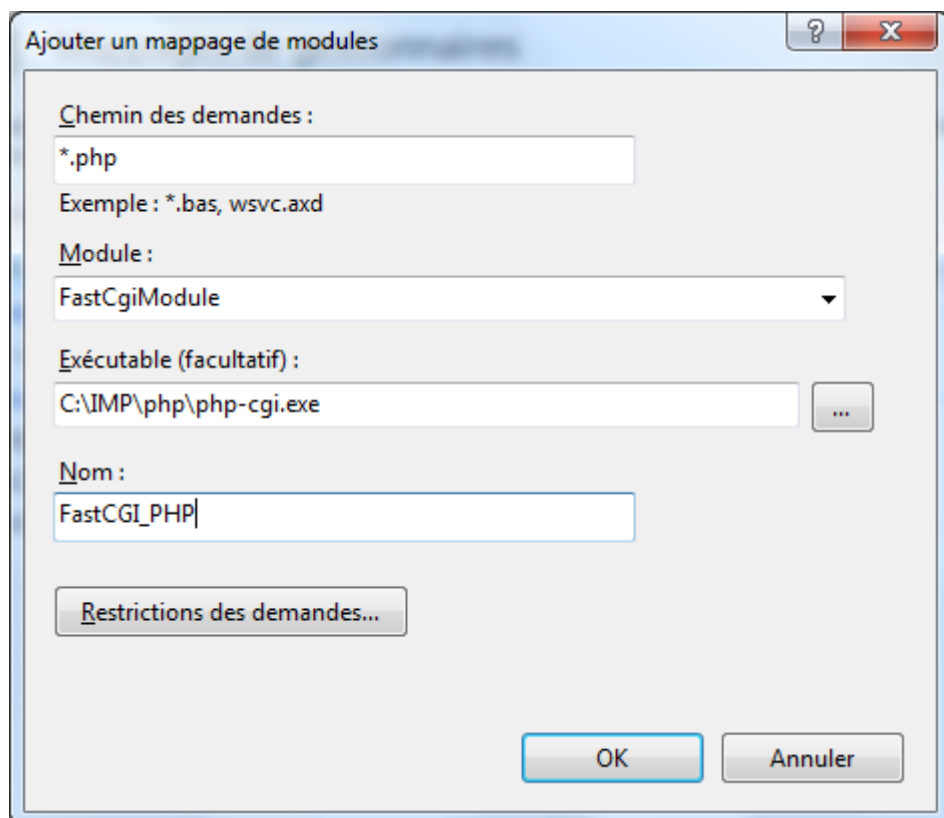
Maintenant, allez dans l'interface **Gestionnaire des services Internet (IIS)** (vous pouvez également appuyer simultanément sur les touches **Windows** et **R** de votre clavier et saisir **inetmgr** pour aller plus vite à l'avenir) puis double-cliquez sur **Mappages de gestionnaires** :



Dans la colonne de droite, cliquez maintenant sur **Ajouter un mappage de modules...**

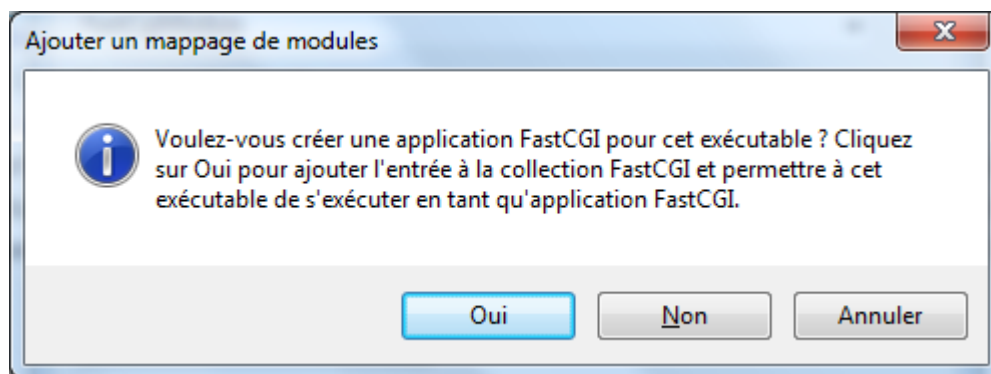


Une nouvelle fenêtre va maintenant s'afficher. Voici les différentes informations à saisir :



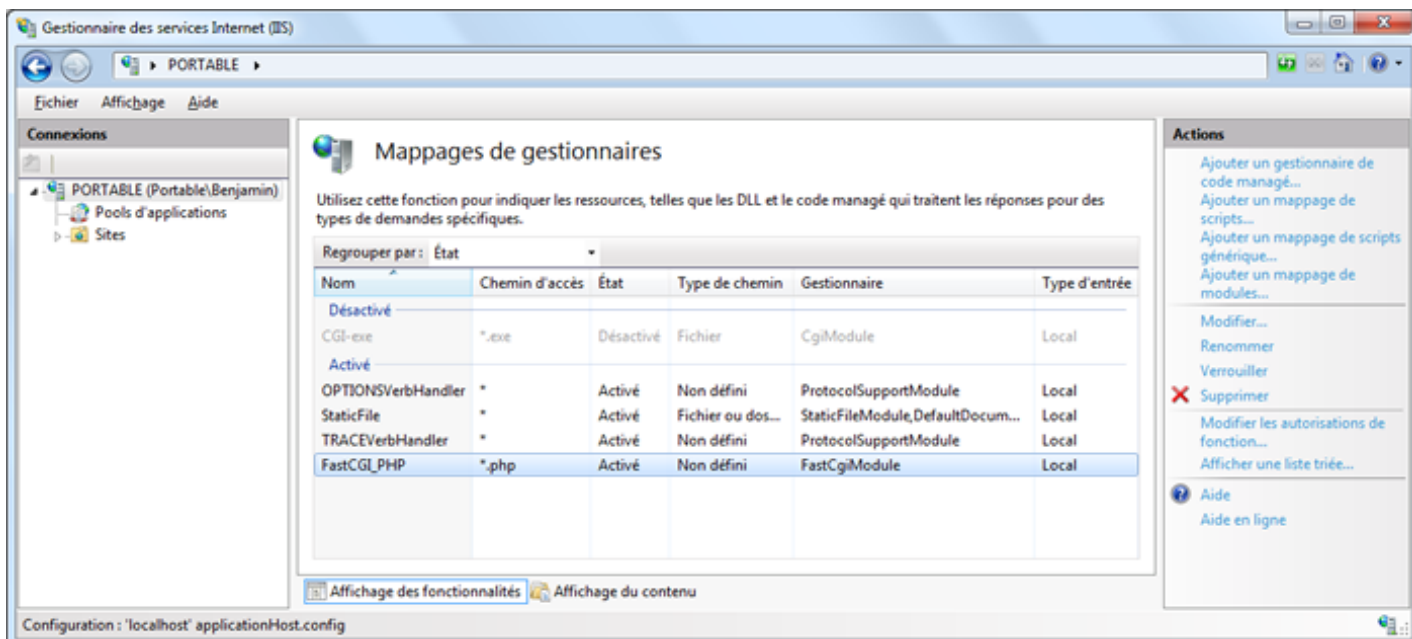
Lors du choix de l'exécutable, la liste des fichiers est filtrée et seuls les fichiers portant l'extension **dll** sont affichés. En bas de la fenêtre, vous pouvez changer pour **.exe** et c'est ce qu'il faudra faire si vous ne souhaitez pas saisir le chemin à la main.

La saisie des différents champs effectuée, cliquez sur le bouton **OK** pour continuer.



Une confirmation vous est demandée, cliquez alors sur le bouton **Oui** pour valider.

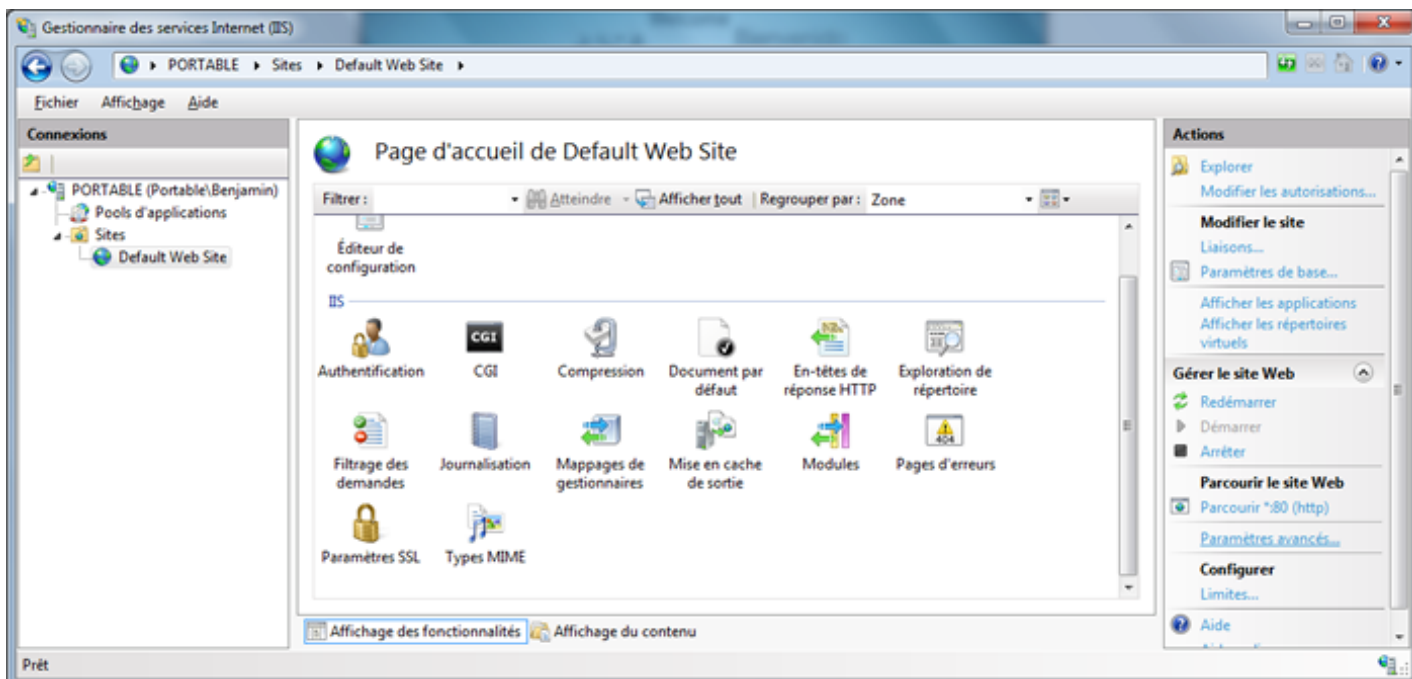
Le nouveau module est maintenant ajouté à la liste des modules actuellement disponibles (ici en surbrillance):



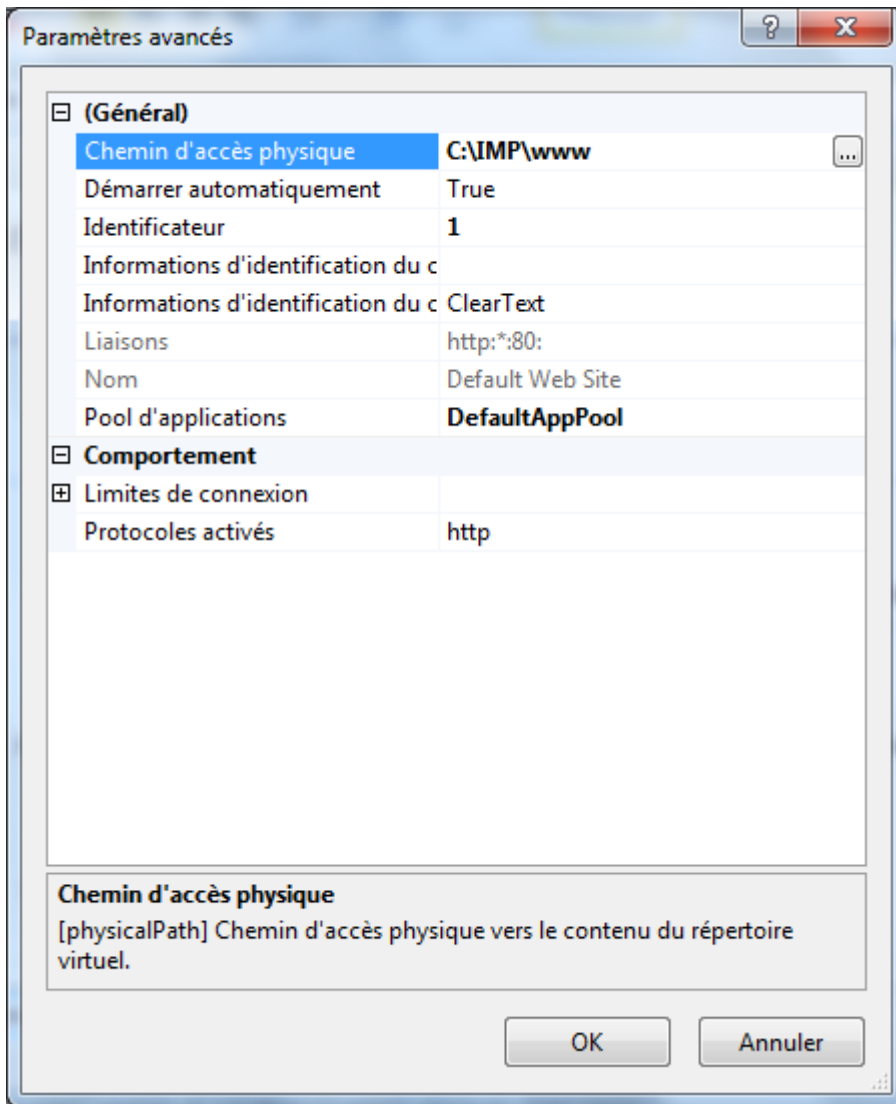
Concrètement, cela signifie qu'à chaque page portant l'extension php appelée via IIS, celle-ci sera alors traitée par PHP en mode (Fast)CGI.

Nous allons maintenant vérifier si c'est bien le cas. Cependant, il faut modifier le répertoire racine de votre site Web.

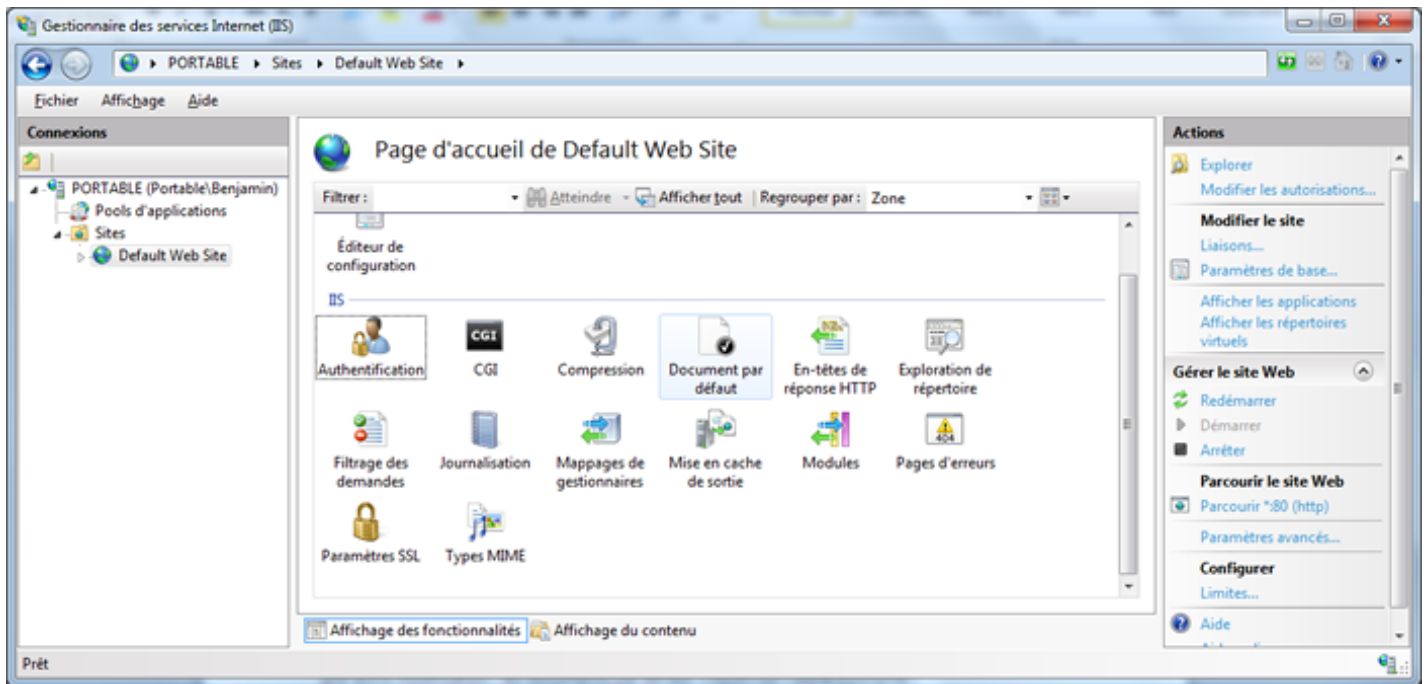
Dans mon cas, je vais utiliser celui créé par défaut par IIS (à savoir **Default Web Site**) pour l'adapter à mes besoins (le port 80 étant bien pratique, je ne vais pas en créer un autre et il me semble inutile de supprimer l'existant pour recréer quasiment le même). Naviguez alors dans la fenêtre de gauche du **Gestionnaire des services Internet (IIS)** jusqu'à **Default Web Site**.



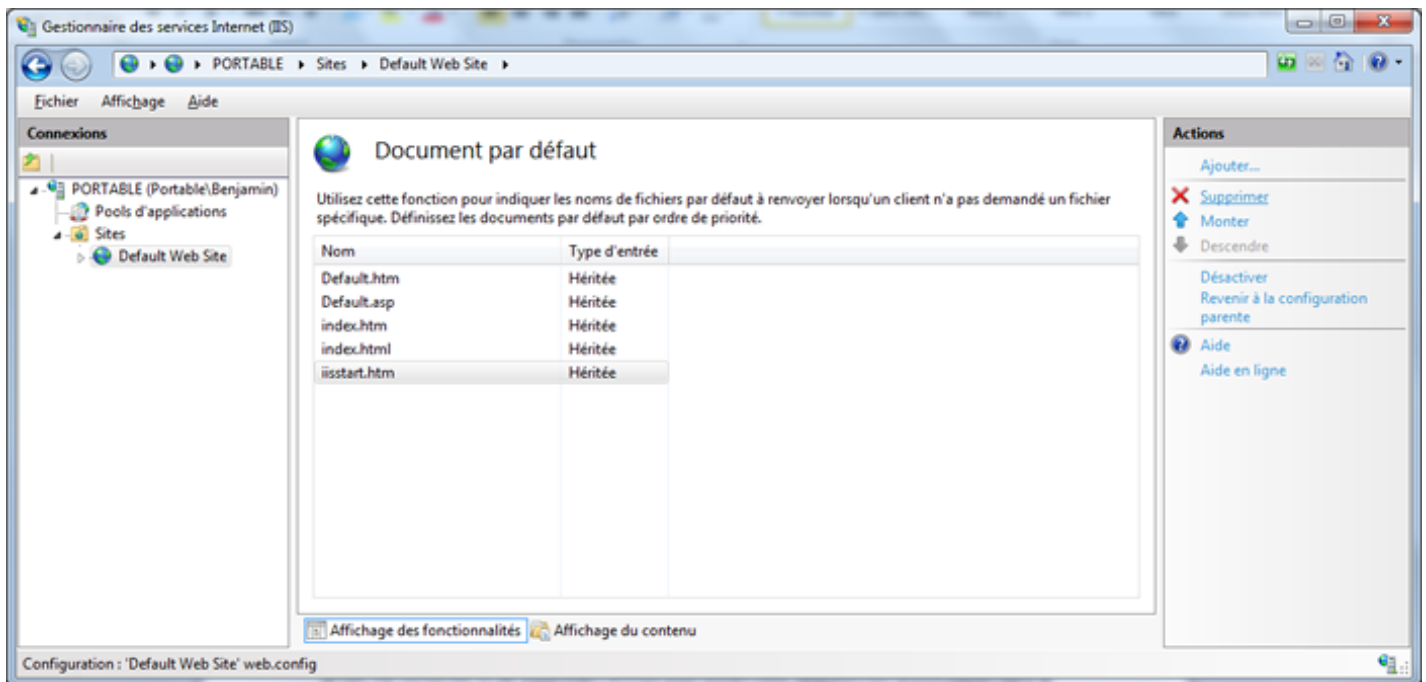
Nous allons donc cliquer, dans le cadre de droite, sur **Paramètres avancés**. Dans la nouvelle fenêtre qui s'affiche, il vous faut maintenant modifier le **Chemin d'accès physique**. Comme expliqué plus haut, j'ai choisi **C:\IMP\www**. Je navigue et valide donc ce répertoire :



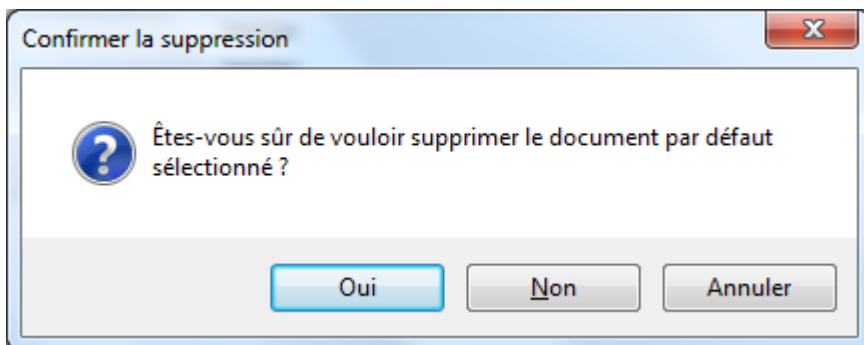
De retour à la console de gestion de votre site Web par défaut, nous allons en profiter pour apporter une autre modification : les documents par défaut. Ces documents sont affichés si aucune page n'est précisée dans l'adresse demandée par le client au serveur (bien souvent ce sont index.htm, index.html, index.php, etc.). Cliquez sur l'icône éponyme :



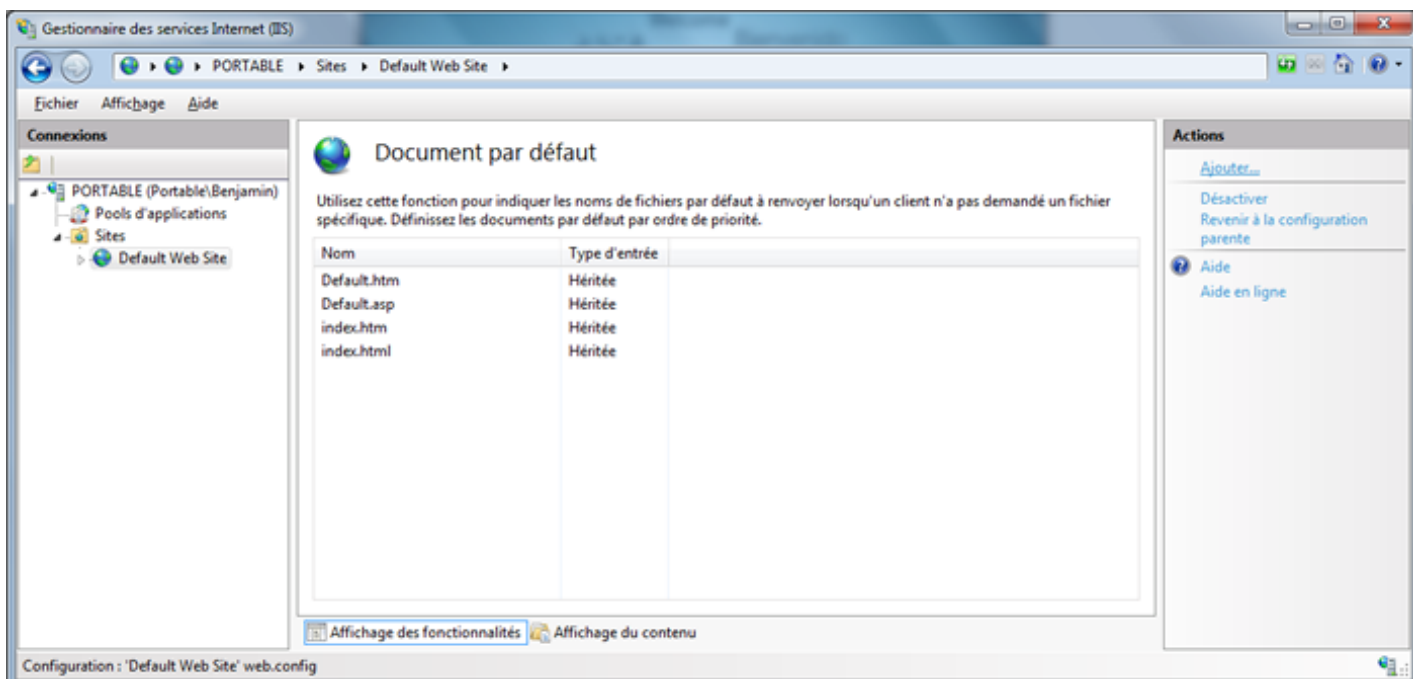
Je vais me contenter ici de supprimer l'entrée pour **iisstart.htm** (page par défaut créée à l'installation de IIS). Sélectionnez-la puis cliquez dans la colonne de droite sur **Supprimer** :



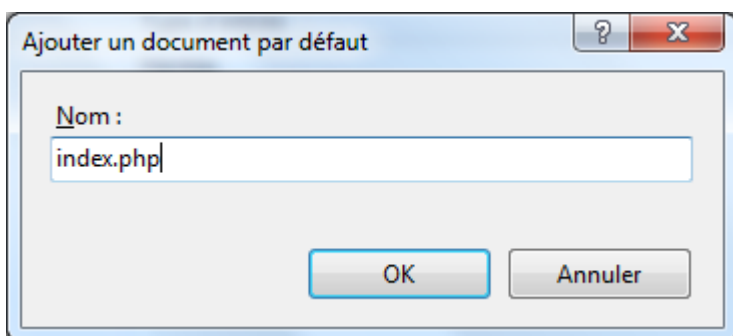
Validez votre choix en cliquant sur **Oui** :



Nous allons maintenant ajouter **index.php** comme page par défaut. Cliquez alors sur **Ajouter...** en haut, dans la colonne de droite :



Vous vous en doutez, dans la fenêtre qui vient de s'ouvrir, il faut saisir **index.php** :



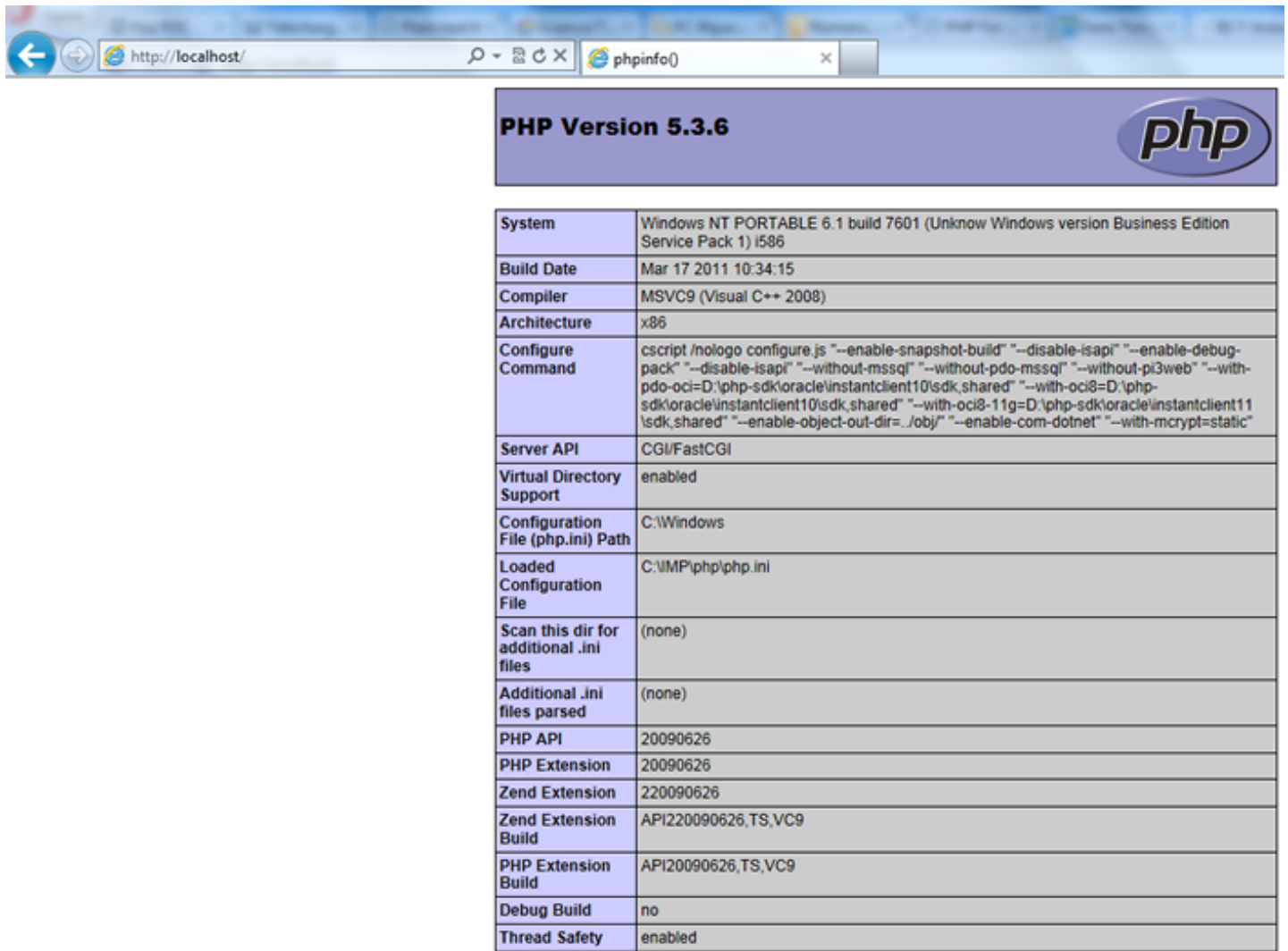
Cliquez sur **OK** pour valider.

Suivant le ou les langages que vous utilisez le plus, il peut être pratique de faire en sorte que la page nommée **index.php** soit prioritaire sur les autres pages par défaut. Si c'est le cas, cliquez sur **index.php** puis autant de fois que nécessaire sur **Monter** dans la colonne de droite.

Ces modifications apportées, créez un fichier **index.php** dans le répertoire de base de votre serveur (celui défini à l'étape d'avant, **C:\IMP\www** pour moi). Editez-le ensuite avec le bloc-notes par exemple et saisissez :

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

C'est maintenant l'heure du moment de vérité ! Si vous avez bien suivi ce tutoriel (en espérant avoir été le plus clair possible), utilisez votre navigateur et rendez-vous à l'adresse **http://localhost**, vous devriez vous trouver devant une page telle que celle-ci :



Comme vous pouvez le voir, nous utilisons donc PHP dans sa version 5.3.6, sous Windows 6.1 (Windows 7 donc) compilé sous Visual C++ 2008 (d'où le VC9 dans le nom de l'archive de PHP).

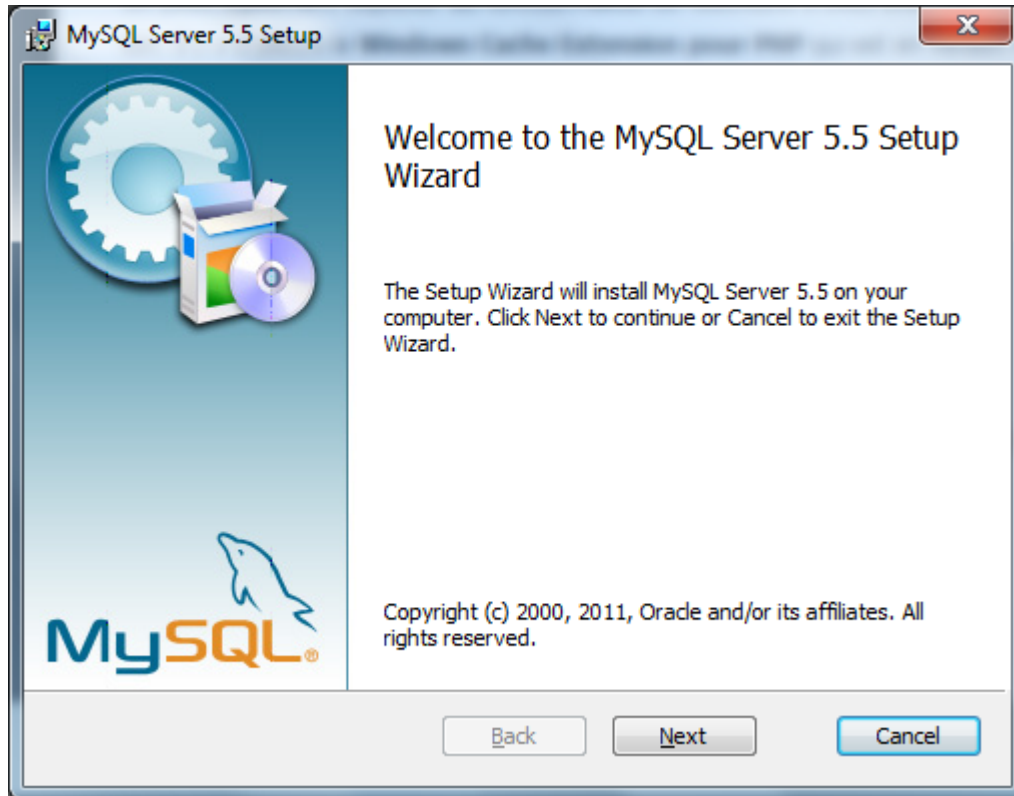
A la ligne **Server API**, vous constaterez que nous utilisons du **CGI/FastCGI**, ce qui est bon signe, vous en conviendrez.

Notre serveur Web commence à prendre forme. Il reste à installer le serveur de base de données MySQL et c'est ce qui va nous occuper dans le prochain chapitre.

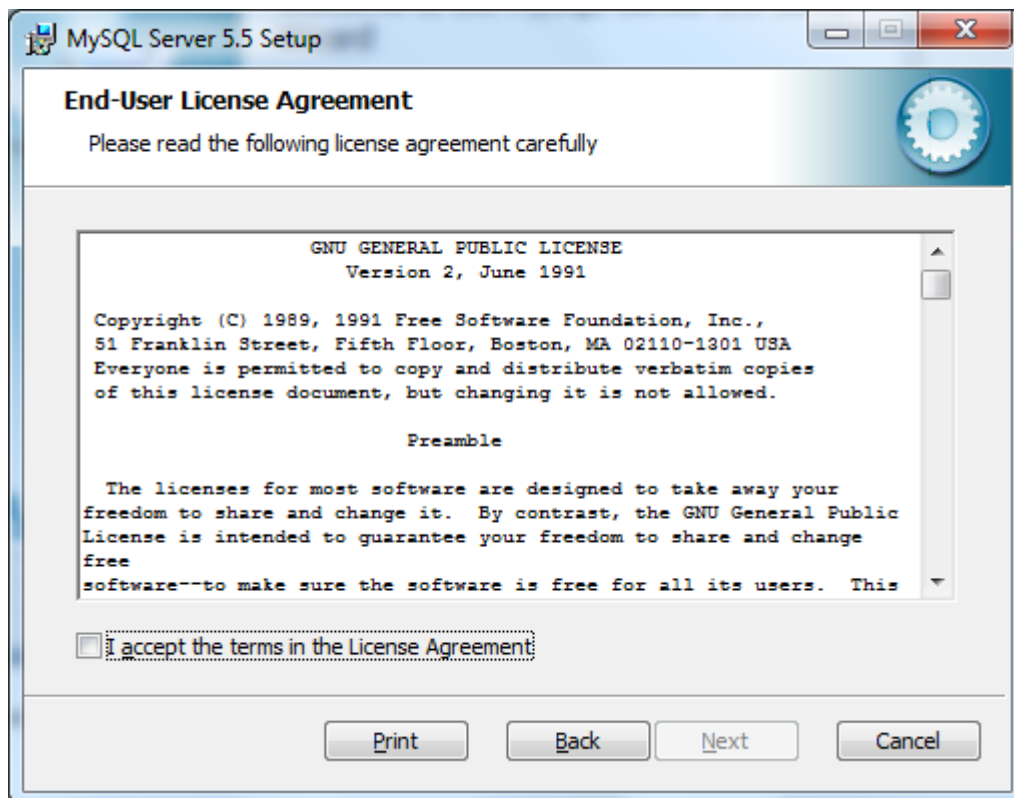
IV - Installation et configuration de MySQL

Dans ce chapitre nous allons voir comment télécharger, installer et configurer MySQL. Pour le télécharger, rien de plus simple : rendez-vous à l'adresse <http://mysql.fr/downloads/mysql/> et choisissez ensuite le paquet qui vous convient le mieux. J'opte en ce qui me concerne pour l'installateur au format MSI en x86/32 bits (mysql-5.5.14-win32.msi donc).

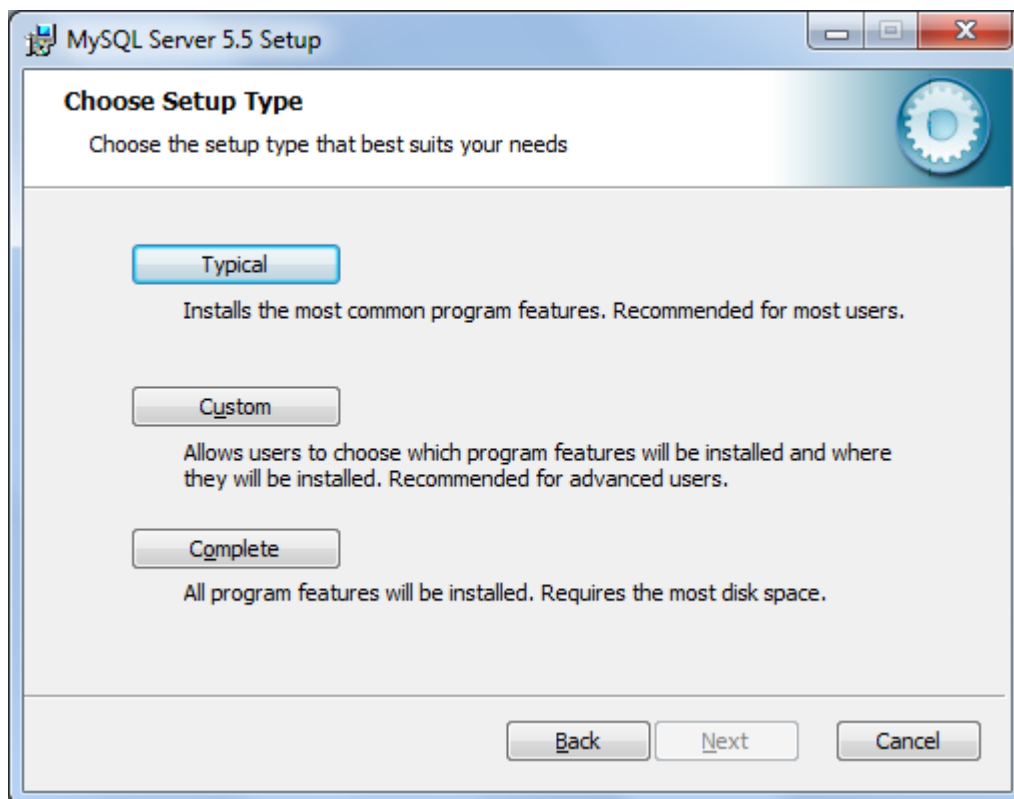
L'installateur téléchargé, double-cliquez dessus pour le lancer. Nous allons faire l'installation pas à pas.



A ce premier écran, cliquez sur **Next**.

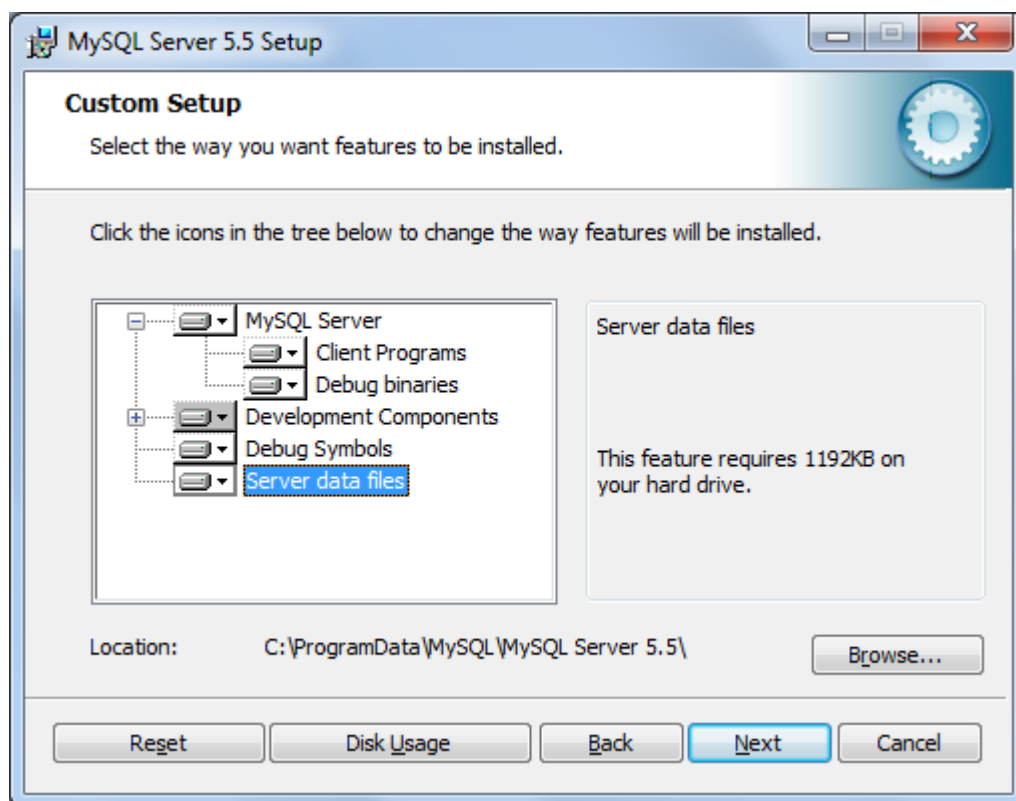
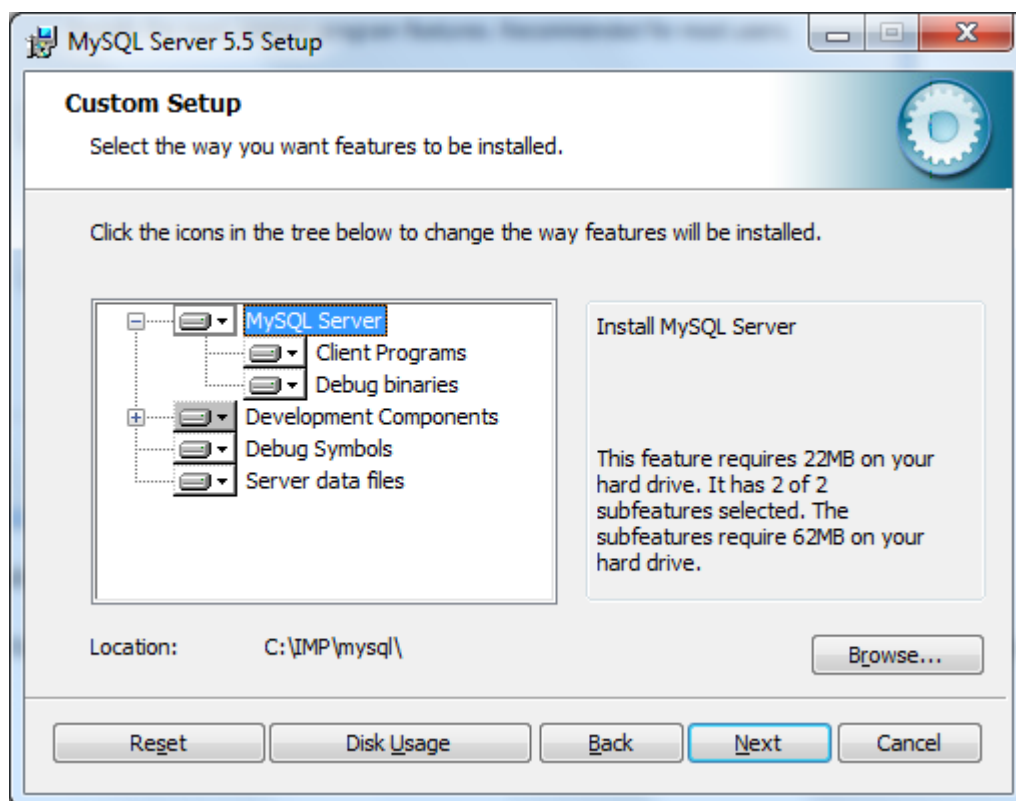


Cochez la case **I accept the terms in the License Agreement** (cela implique que vous lisez et acceptiez les conditions) puis cliquez sur **Next**, une fois encore.

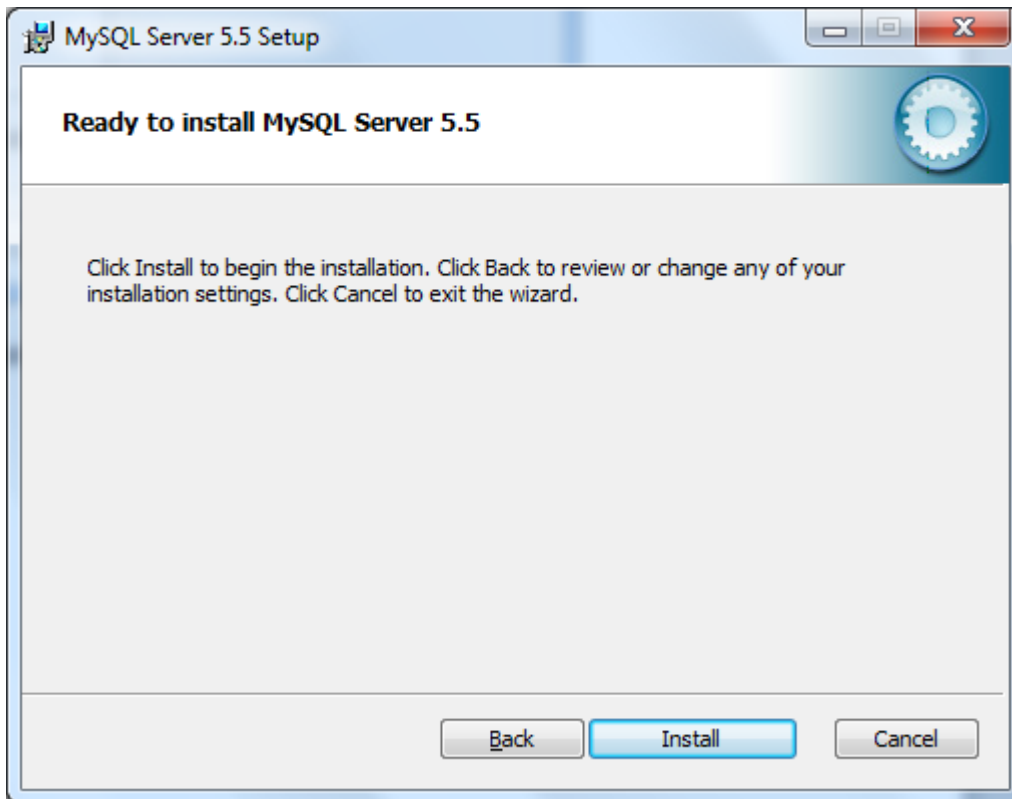


Ici, nous allons choisir **Custom** afin de réaliser une installation personnalisée.

Je vais installer MySQL Server dans **C:\IMP\mysql**, je laisse **Server data files** dans le répertoire par défaut :



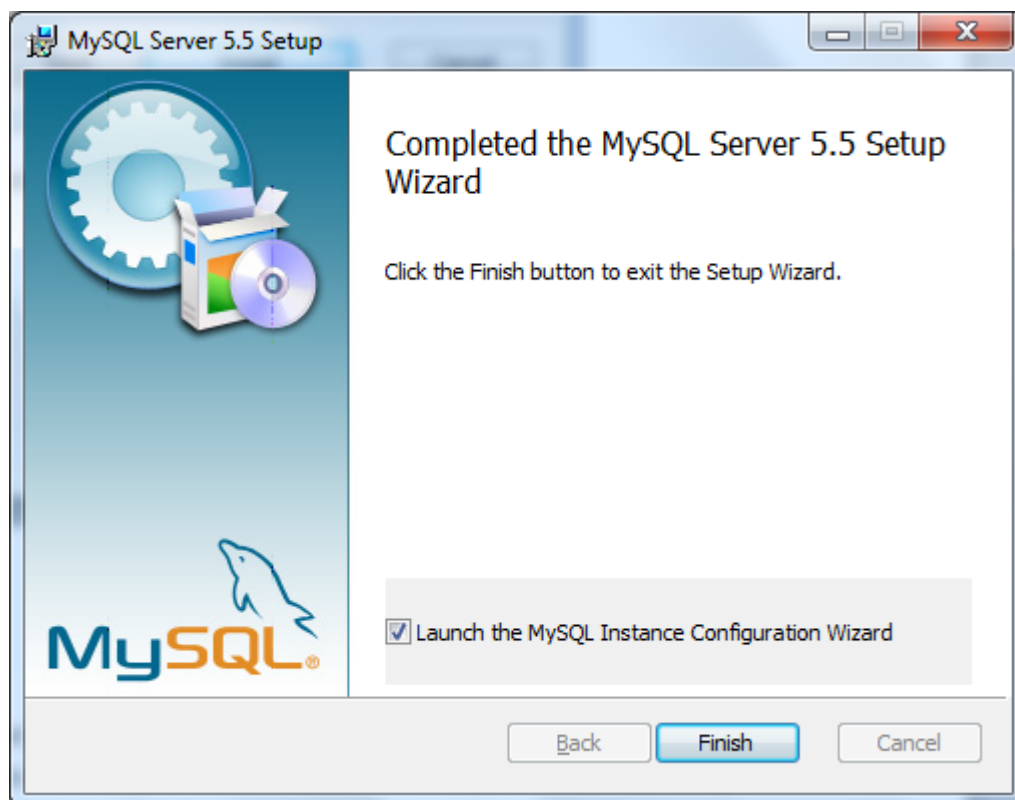
On clique sur **Next** pour passer à l'étape suivante.



Cliquez sur **Install** pour lancer l'installation de MySQL.

Si l'UAC (User Account Control, permet l'élévation de privilège) est activé sur votre ordinateur, une confirmation vous sera alors demandée, MySQL souhaitant apporter des modifications à votre système (rien de plus normal pour une installation).

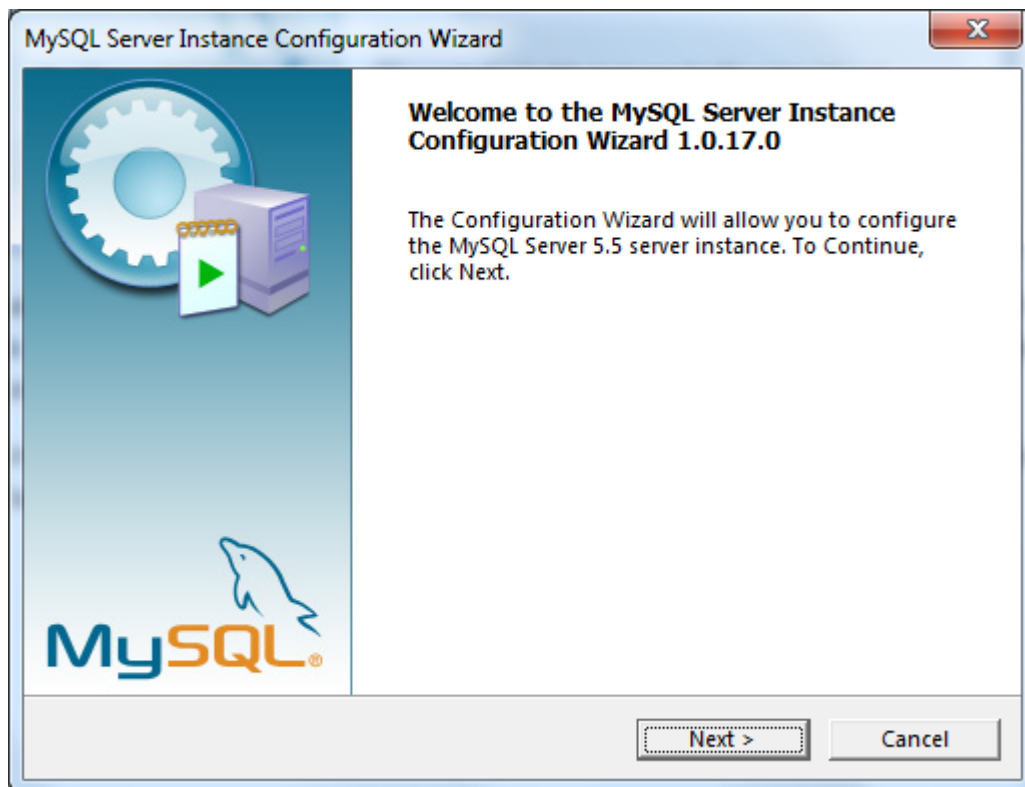
L'installation faite, cliquez deux fois sur **Next** pour passer à la fin de l'installation :



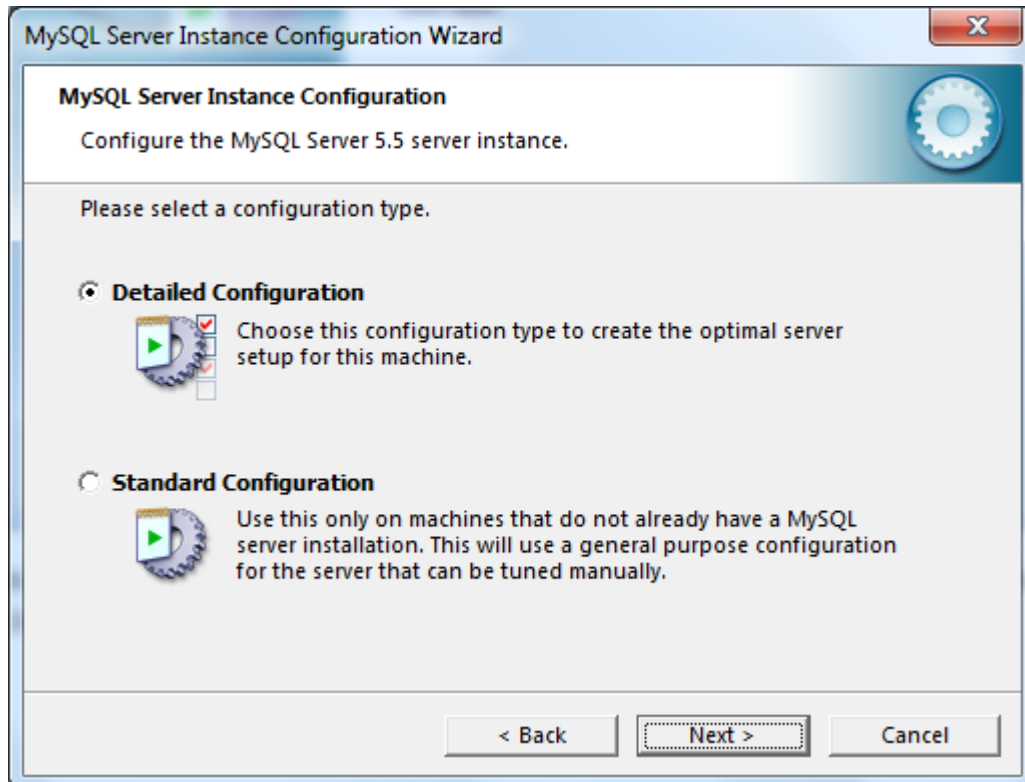
Laissez la case **Launch the MySQL Instance Configuration Wizard** cochée afin de passer directement à la configuration du serveur MySQL. Cliquez sur **Finish**, l'installation étant terminée. Passons à la configuration du serveur.

Une nouvelle fois, une confirmation vous sera demandée par l'UAC. Cliquez sur Oui pour démarrer l'assistant de configuration.

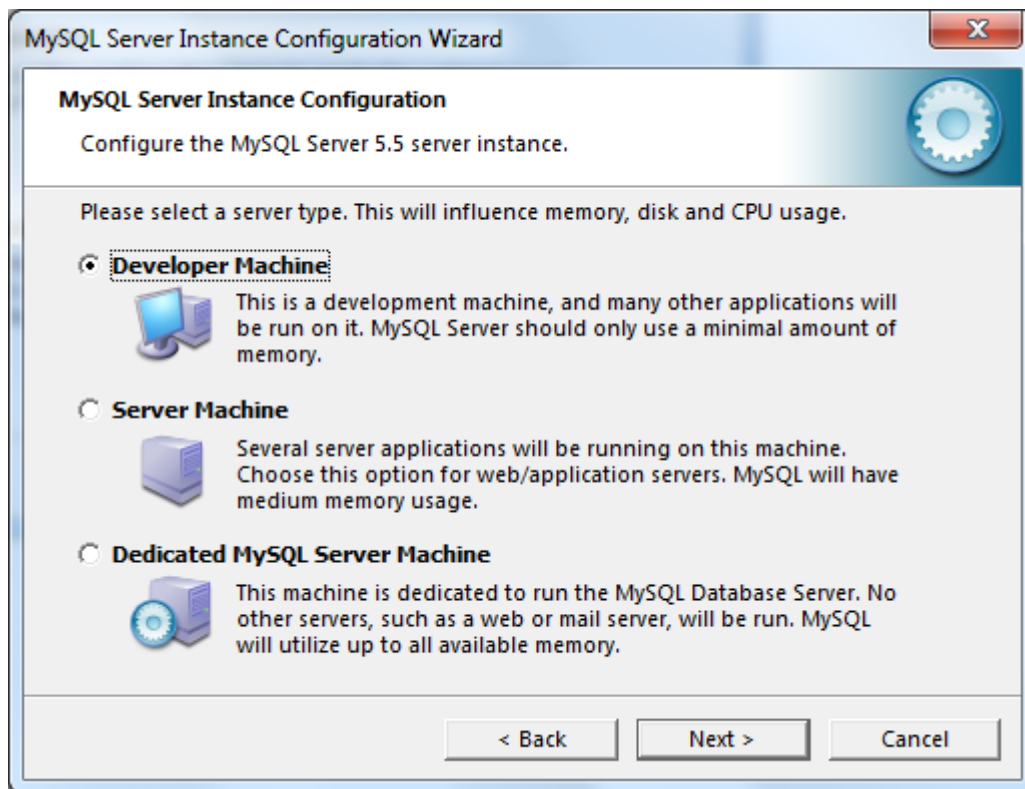
L'assistant lancé, cliquez sur **Next** pour débiter :



Nous allons faire une configuration détaillée (Detailed configuration), l'option par défaut donc. Cliquez sur Next pour l'étape suivante :

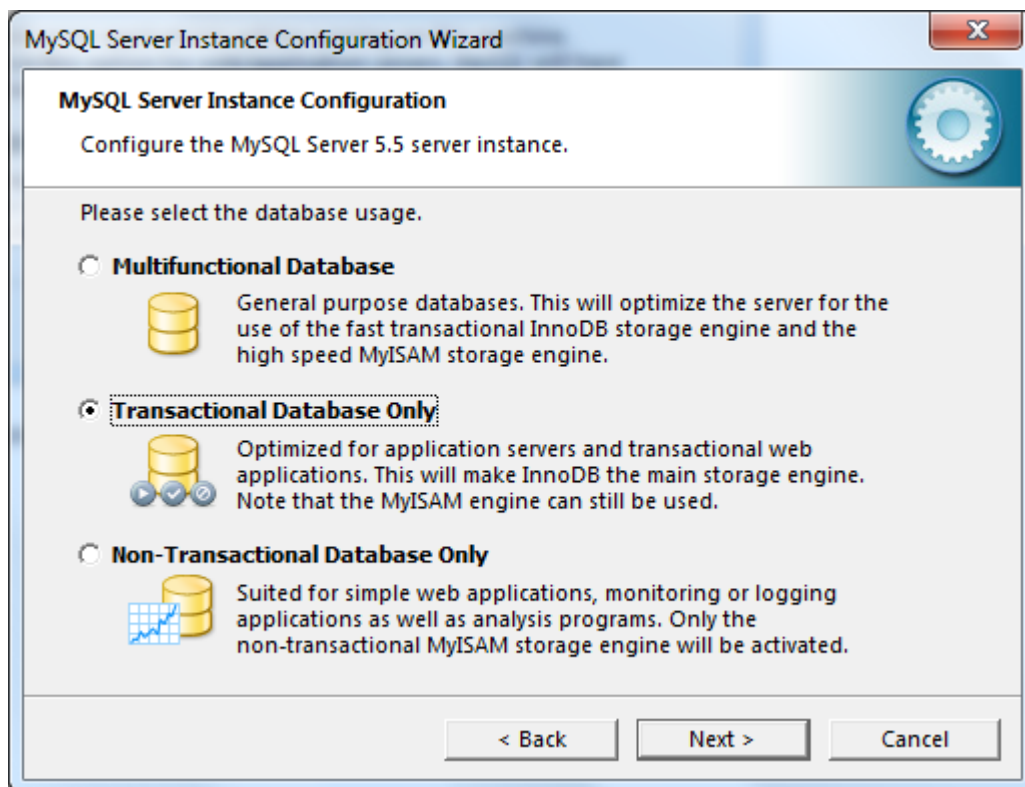


Une fois n'est pas coutume, l'option par défaut est très bien. Dans notre cas, il s'agit d'une machine de développement, inutile d'opter pour les autres options dans notre cas :

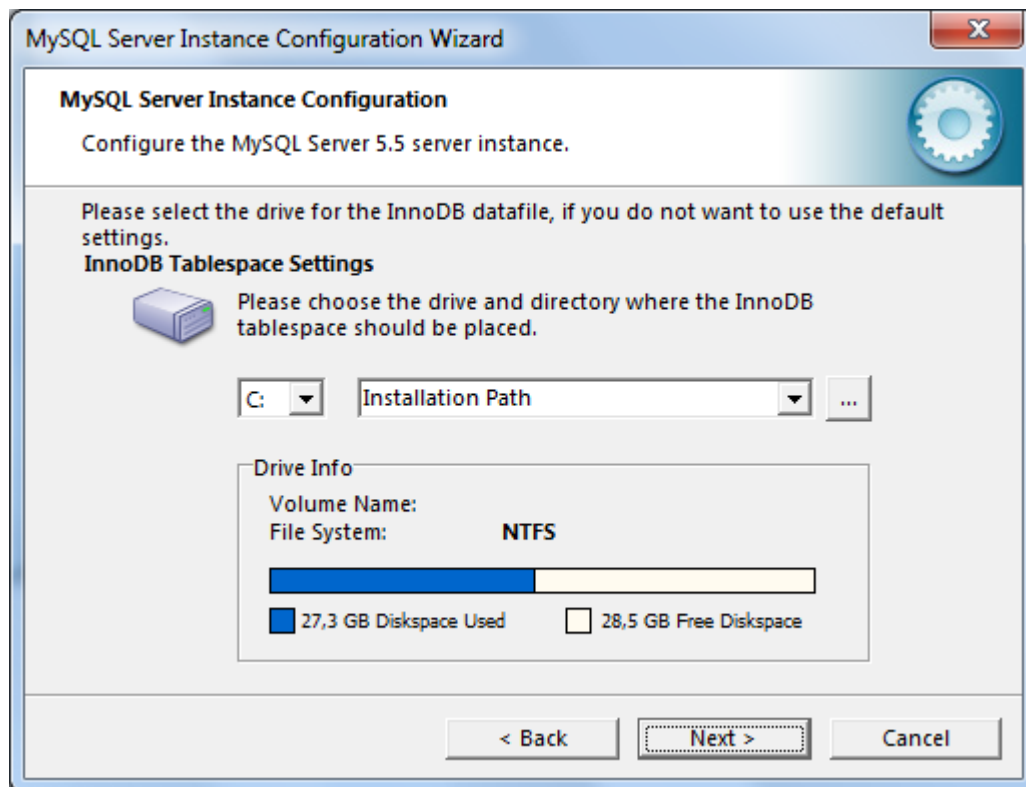


Cliquez alors sur Next pour passer à la suite.

Dans le cas présent, j'aime avoir InnoDB et MyISAM, j'opte donc pour la seconde option. Pour un petit serveur de test, vous pouvez opter pour la dernière case à cocher, à savoir **Non-transactional Database Only**. InnoDB ne sera alors pas installé et disponible mais ça ne gêne en rien l'utilisation courante de votre base de données.

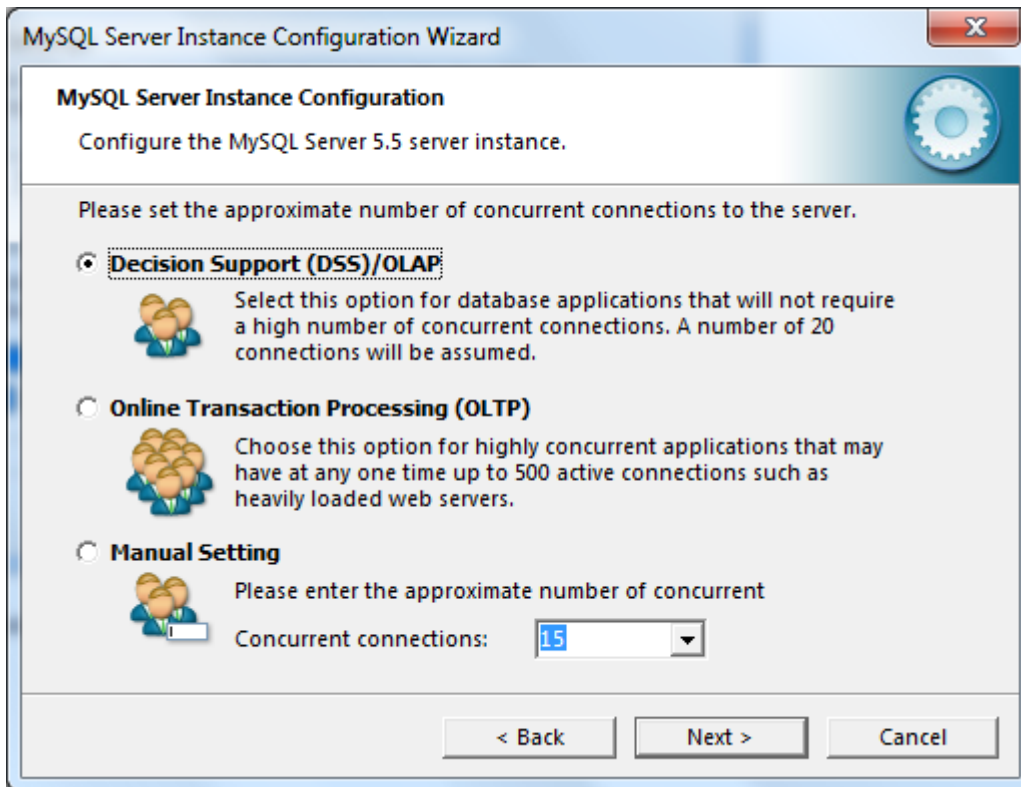


Cliquez sur **Next** pour l'étape d'après.

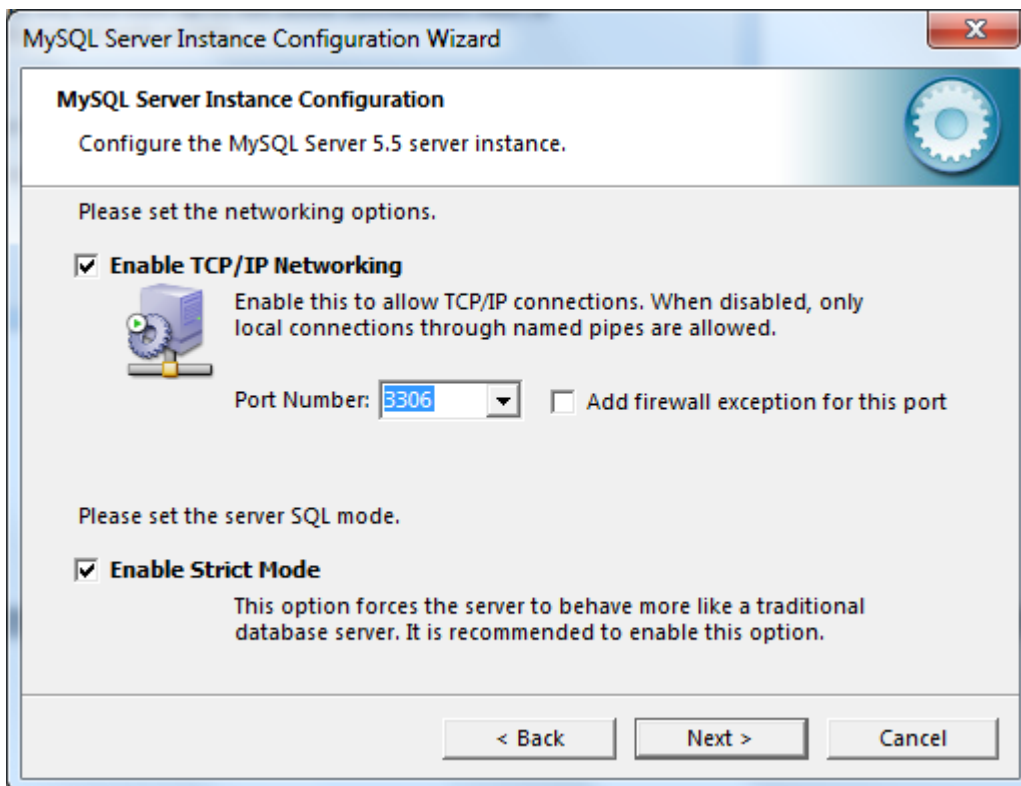


Laissez ici le répertoire par défaut (basé sur le répertoire d'installation de MySQL) si vous avez choisi d'installer InnoDB puis cliquez sur **Next** pour passer à la suite.

Une fois encore, l'option choisie par défaut va très bien, je ne développe pas de gros projets avec des grosses bases de données. Va pour le premier choix donc :



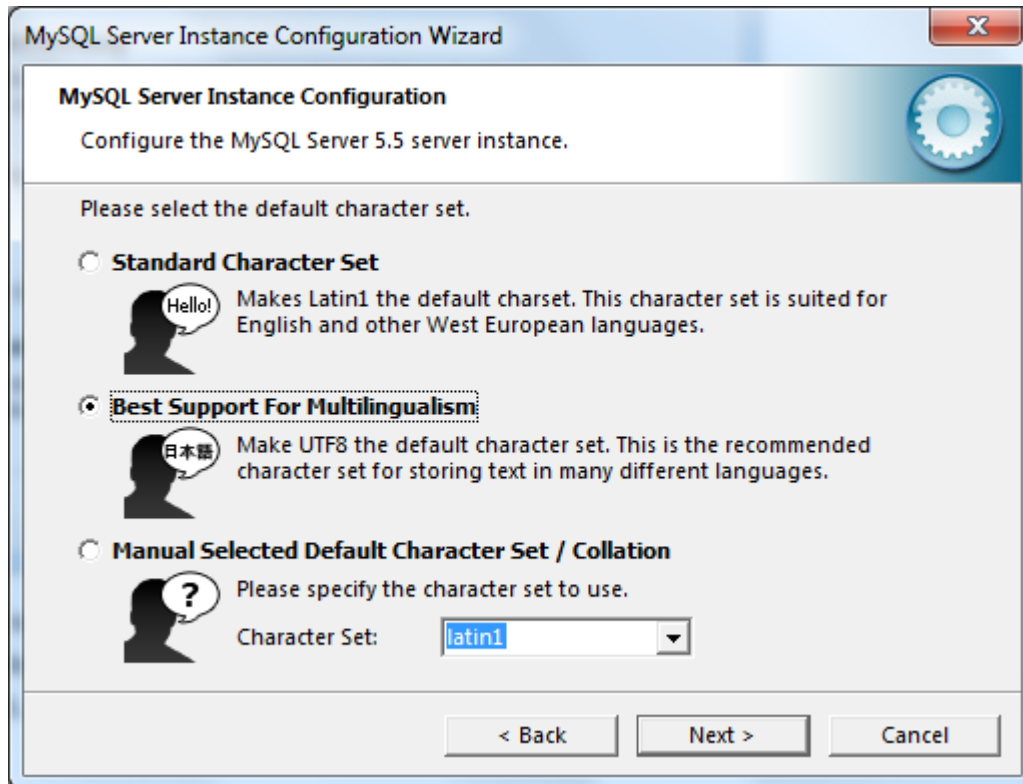
Cliquez sur **Next**.



Là encore, les paramètres par défaut sont biens. Notez que je n'ajoute pas d'exception au pare-feu Windows, je n'ai pas de connexion depuis un autre poste/PC pour le moment mais garde cette éventualité dans la configuration de MySQL (d'où le fait de garder la première case cochée).

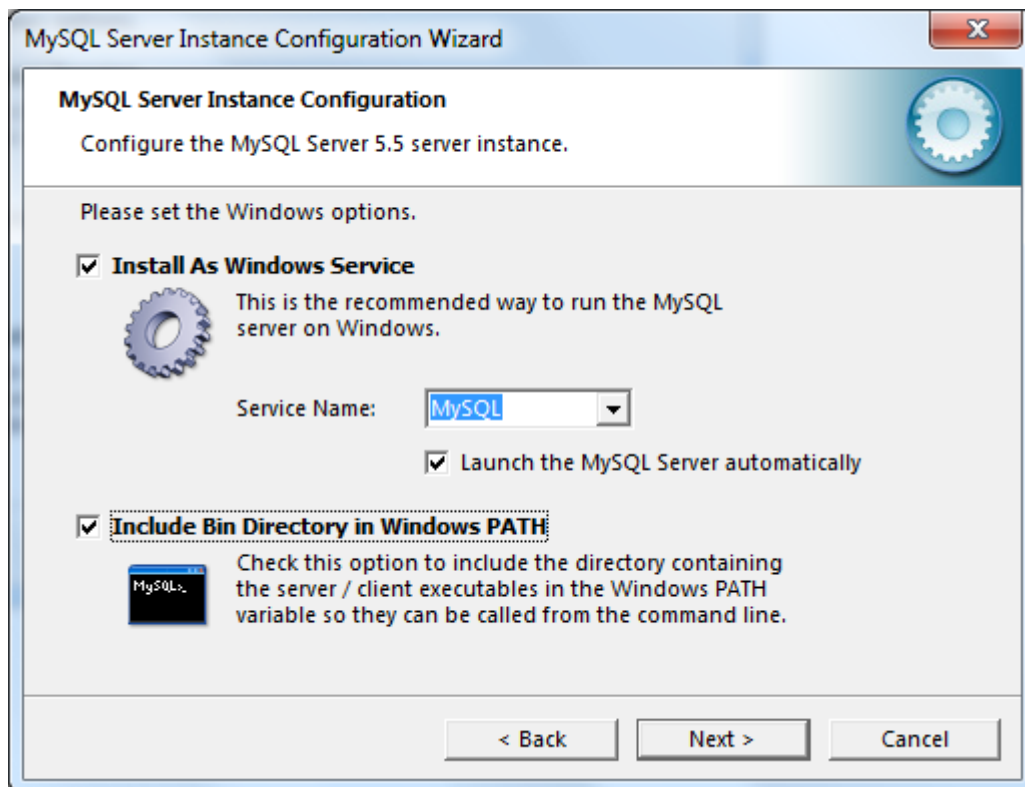
Cliquez sur **Next**.

Pour le choix du jeu de caractères, je sais que je vais avoir différents caractères non latin à stocker dans mes bases de données et opte alors pour de l'utf8. Dans la grande majorité des cas, le choix par défaut suffira. Vous pouvez néanmoins relancer l'assistant de configuration de MySQL à tout moment. Ce dernier est dans le dossier **C:\IMP\mysql\bin** et se nomme **MySQLInstanceConfig.exe**.

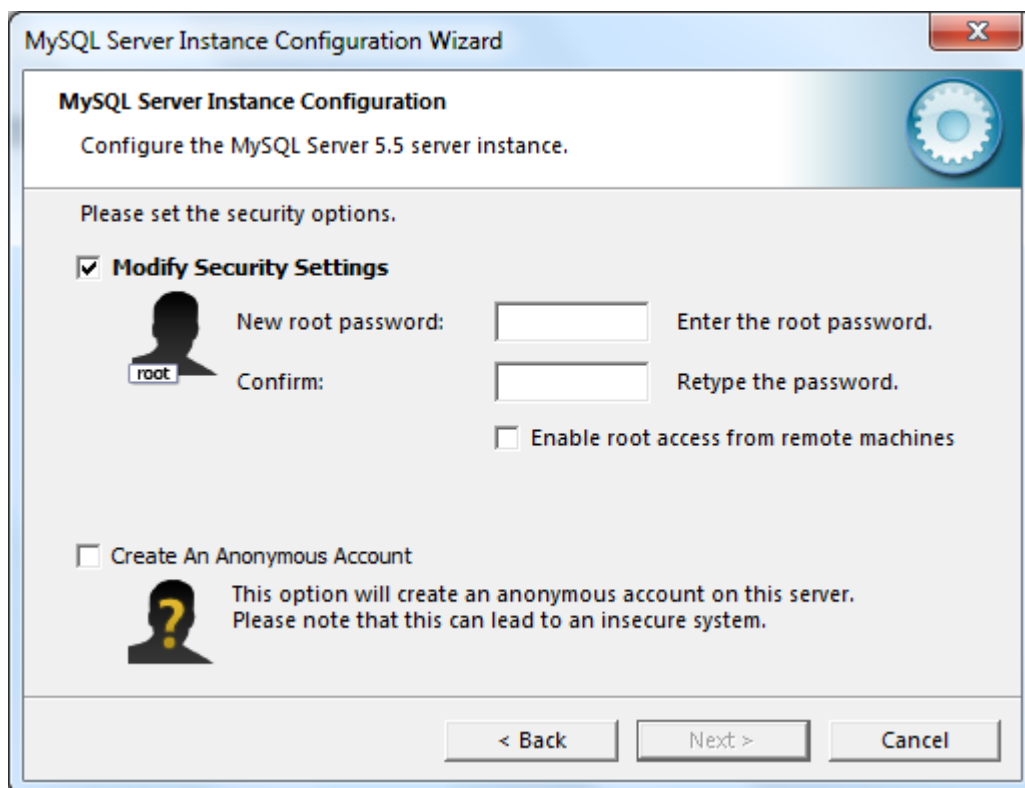


Cliquez sur **Next**.

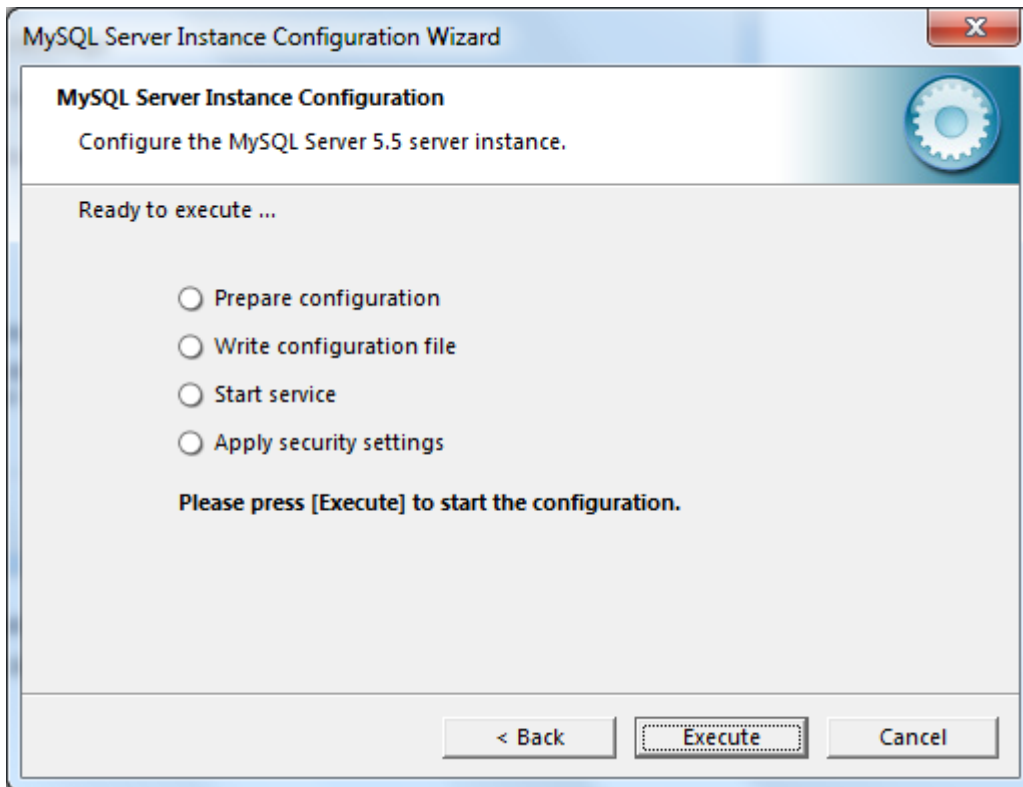
J'installe MySQL comme service, pour la simple raison que c'est transparent : le serveur démarre au lancement de Windows, vous n'avez donc rien à lancer. Ça me convient très bien de cette façon. Je coche par contre la seconde case, il peut être pratique et intéressant d'ajouter le répertoire de MySQL contenant divers outils (mysqldump et consorts) aux variables d'environnement de Windows. Cela signifie que vous pouvez appeler ces exécutables depuis une invite de commande DOS sans avoir à naviguer jusqu'au répertoire dans lequel sont ses exécutables. Ceci prend tout sens si vous prévoyez d'utiliser l'outil mysql.exe ou mysqldump.exe depuis un script batch utilisé via une tâche planifiée (par exemple).



Cliquez sur **Next**.

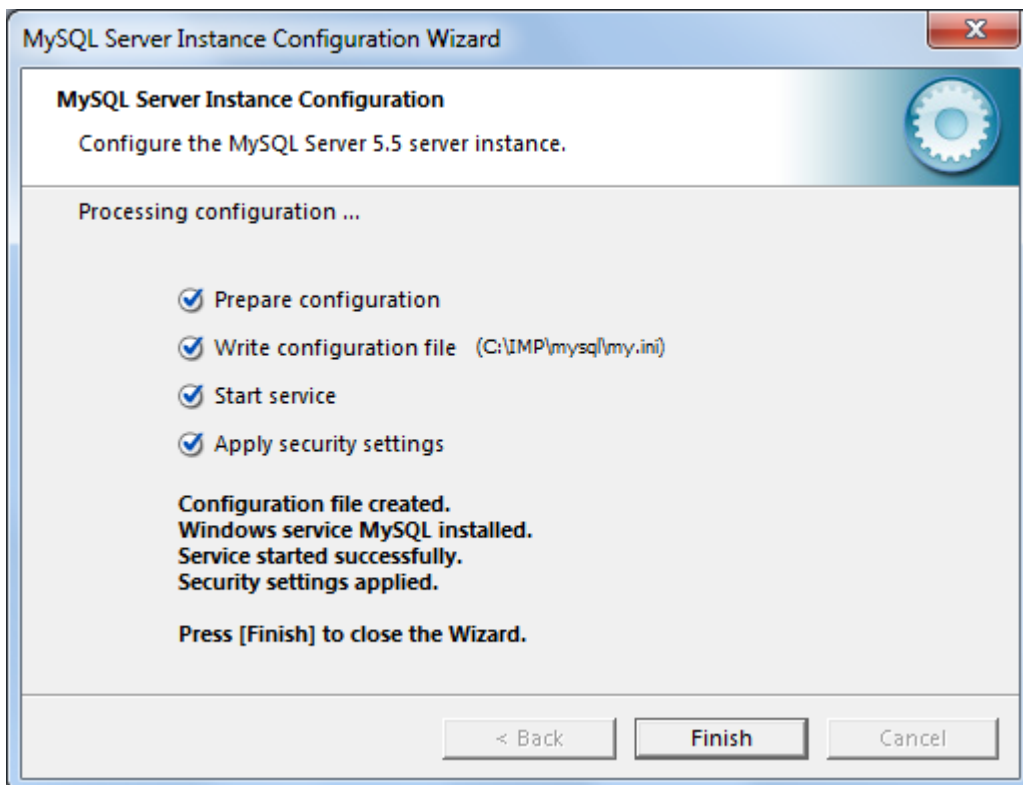


Saisissez ici le mot de passe du compte root. En ce qui me concerne, je ne souhaite pas autoriser les accès à distance depuis le compte root ni créer un compte "anonyme". Cliquez sur **Next** une fois les informations saisies.



La configuration terminée, il suffit maintenant de cliquer sur **Execute** pour valider les paramètres.

Si tout s'est bien passé (aucune raison que ça se passe mal), un message vous l'indique :



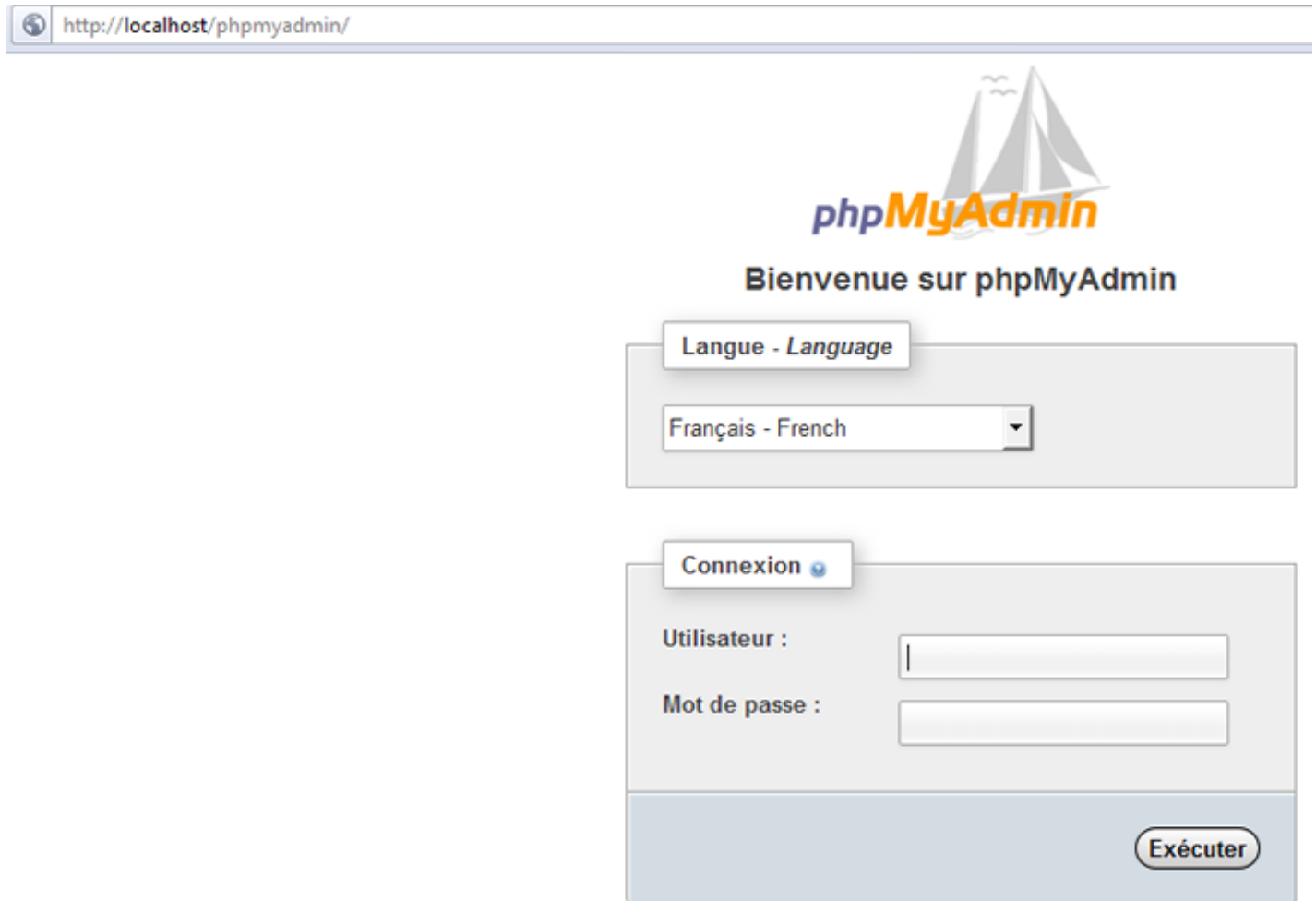
Cliquez sur **Finish** pour terminer la configuration.

Vous voilà désormais avec un serveur de développement basé sur Windows, avec IIS, MySQL et PHP. Cependant, il manque quelques petites choses pratiques pour en profiter pleinement et c'est l'objet des différents chapitres suivants.

V - Installation et configuration de PHPMyAdmin

PHPMyAdmin, vous le savez peut-être déjà, vous permet de gérer vos bases de données. L'installer vous permettra de le faire depuis une interface Web intuitive. Rendez-vous à l'adresse suivante pour le télécharger dans sa dernière version : http://www.phpmyadmin.net/home_page/downloads.php.

Dans mon cas, phpMyAdmin-3.4.3.2-all-languages.7z me semble la plus adaptée (version avec une traduction française et la plus légère à télécharger). Le téléchargement effectué, j'extrais le contenu de l'archive à la racine de mon serveur (C:\IMP\www) dans un dossier nommé phpmyadmin. De cette manière, PHPMyAdmin sera directement accessible à l'adresse <http://localhost/phpmyadmin> comme vous pouvez le voir ci-dessous :



Simple comme bonjour.

Cependant quelques petites modifications sont à apporter dans la configuration de PHPMyAdmin. Dans le dossier où vous avez extrait PHPMyAdmin, il y a un fichier nommé **config.sample.inc.php**. Pour commencer, renommez-le en **config.inc.php**.

Ouvrez-le ensuite avec un éditeur de texte (autre que le bloc-notes, à cause des retours à la ligne utilisés).

Modifiez `$cfg['blowfish_secret'] = ''`; Il vous faut saisir une chaîne de caractères (ce que bon vous semble) entre les deux apostrophes (chiffres, lettres majuscules, minuscules, caractères spéciaux etc). Il s'agit ici de créer l'équivalent d'une chaîne de caractère aléatoire.

L'extension MySQLi étant disponible, autant l'utiliser. Changez `$cfg['Servers'][$i]['extension'] = 'mysqli'`; pour `$cfg['Servers'][$i]['extension'] = 'mysqli'`;

En dessous de cette ligne, vous pouvez ajouter celle-ci :

```
$cfg['Servers'][$i]['hide_db'] = '(information_schema|mysql|performance_schema|phpmyadmin|test)';
```

Cette ligne permet de cacher les tables mentionnées dans PHPMyAdmin. Ces tables cachées, vous ne vous retrouvez qu'avec vos tables et non plus les tables "système".

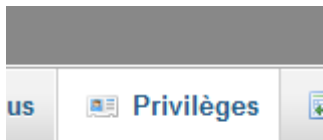
J'ajoute à la suite de `$cfg['SaveDir'] = '';` ces quelques lignes :

```
$cfg['Lang'] = 'fr-utf-8';  
$cfg['ThemeManager'] = FALSE;  
$cfg['ThemeDefault'] = 'original';  
$cfg['ThemePerServer'] = FALSE;  
$cfg['DefaultCharset'] = 'utf-8';  
$cfg['DefaultConnectionCollation'] = 'utf8_bin';
```

Pour faire simple, elles permettent de ne plus avoir la liste déroulante des langues disponibles à la connexion (je suis le seul à utiliser mon PC, donc seul le français me sied). J'ai choisi le thème **pmahomme** par défaut, nul besoin d'un autre et de changer et l'**utf-8** comme jeu de caractères.

Maintenant que PHPMyAdmin est configuré, on va créer un nouvel utilisateur avec son identifiant et sa table (inutile de disposer du compte **root**, même en local).

Allez dans Privilèges, dans le menu en haut de la page :



Cliquez ensuite sur Ajouter un utilisateur :



Et on saisit les différentes informations nécessaires :

Ajouter un utilisateur ✕

Information pour la connexion

Nom d'utilisateur: Entrez une valeur: ▾ benjamin

Client: Local ▾ localhost 1

Mot de passe: Entrez une valeur: ▾

Entrer à nouveau:

Générer un mot de passe: Générer

Base de données pour cet utilisateur

Aucune

Créer une base portant son nom et donner à cet utilisateur tous les privilèges sur cette base

Donner les privilèges passepartout (utilisateur_%)

Privilèges globaux (Tout cocher / Tout décocher)

Annuler
Créer un compte d'utilisateur

Pour récapituler : je créé ici un utilisateur nommé **benjamin** qui pourra se connecter au serveur MySQL local (**localhost**) avec un mot de passe donnée et il disposera d'une base dont le nom sera identique à son nom d'utilisateur (ici **benjamin**).

Maintenant que j'ai mes identifiants de connexion à MySQL, je peux apporter une modification importante : empêcher l'accès du compte root via PHPMyAdmin. En effet, il est inutile de laisser faire cette pratique (volontaire ou non). Pour ce faire, il suffit d'ajouter une ligne dans le fichier de configuration de PHPMyAdmin :

```
<b>$cfg['Servers'][$i]['AllowRoot'] = FALSE;</b>
```

Vous pouvez l'ajouter après `$cfg['Servers'][$i]['AllowNoPassword'] = false;`.

Enregistrez le fichier. Si vous tentez maintenant de vous connecter via PHPMyAdmin via votre compte root, un message d'alerte sera affiché :

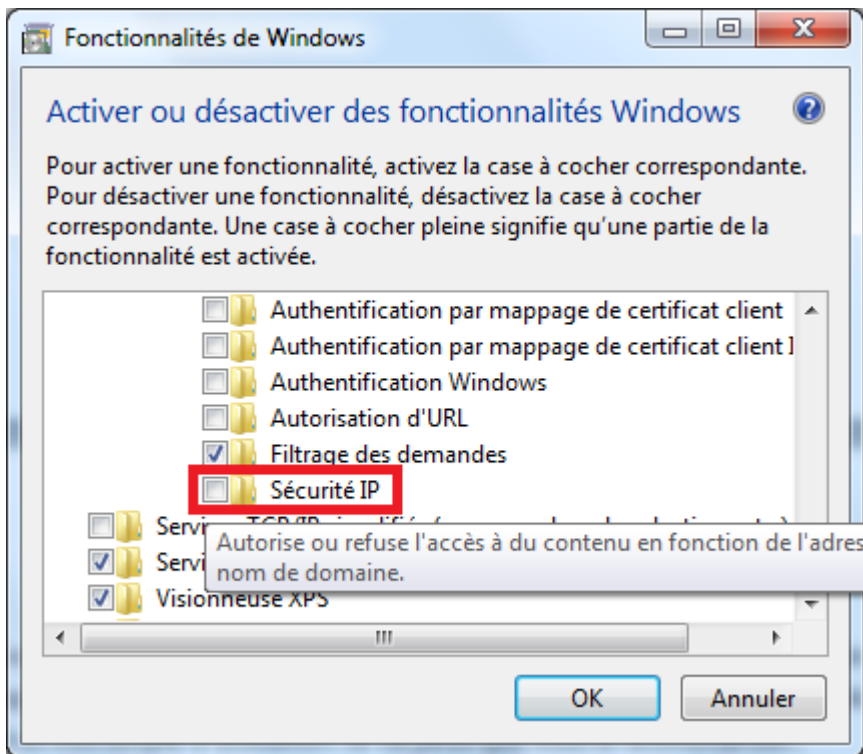
! **Accès refusé**

VI - Filtrage des accès à votre site Web

Vous pouvez, si vous le souhaitez, autoriser ou refuser l'accès à certaines ressources de votre site via un module complémentaire disponible pour IIS. Par défaut, ce dernier n'est pas installé. Il est en tout cas très bien fait et très simple d'utilisation. Je ne peux que vous le recommander si vous pensez en avoir usage.

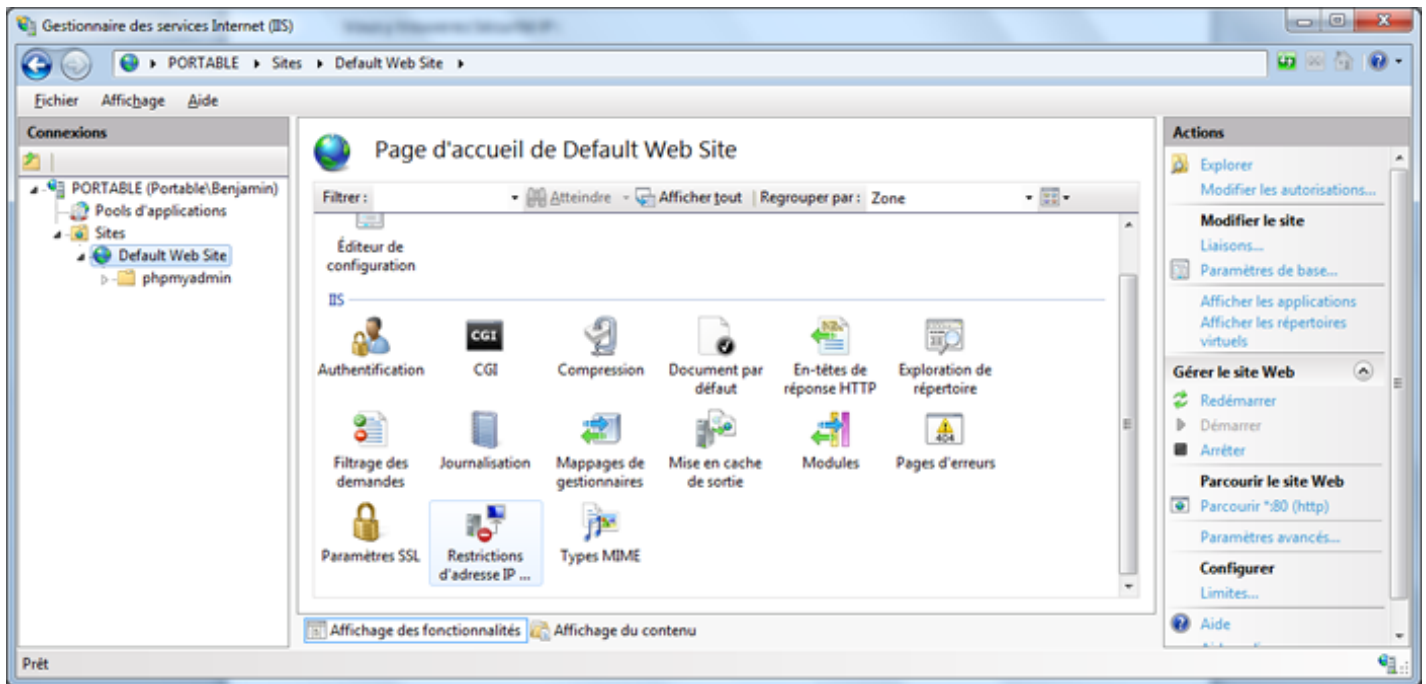
Pour l'installer, rendez-vous dans le **Panneau de configuration**, puis allez dans **Programmes** et choisissez enfin **Activer ou désactiver des fonctionnalités Windows**. Descendez dans la liste jusqu'à trouver **Services Internet (IIS)**. Cliquez sur le **+** pour afficher les options disponibles. Faites de même pour **Services World Wide Web**. Développez la liste des options offertes dans **Sécurité**.

Vous y trouverez alors **Sécurité IP** :



Cochez la case associée et cliquez sur **OK** pour installer le module.

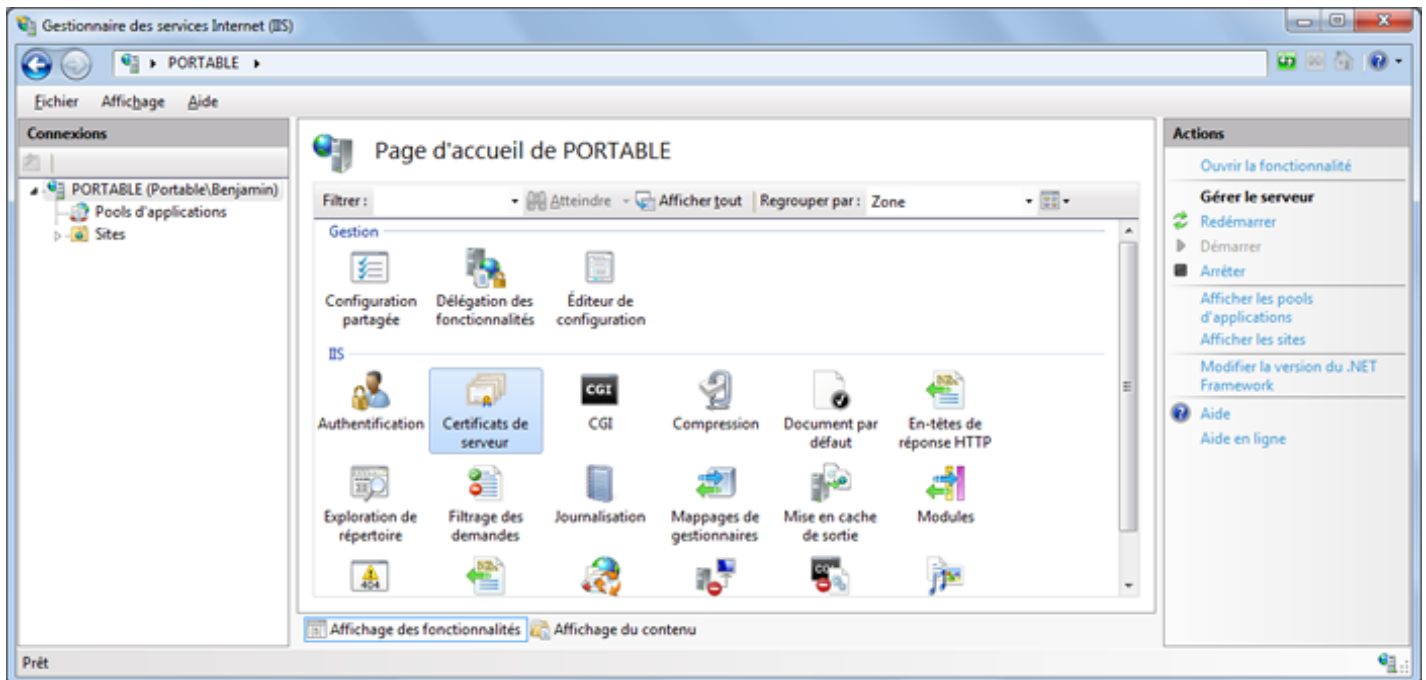
L'installation terminée, fermez le **Gestionnaire de services Internet (IIS)** puis ouvrez-le à nouveau. Vous devriez y trouver une nouvelle icône intitulée **Restriction d'adresse IP et de domaine** :



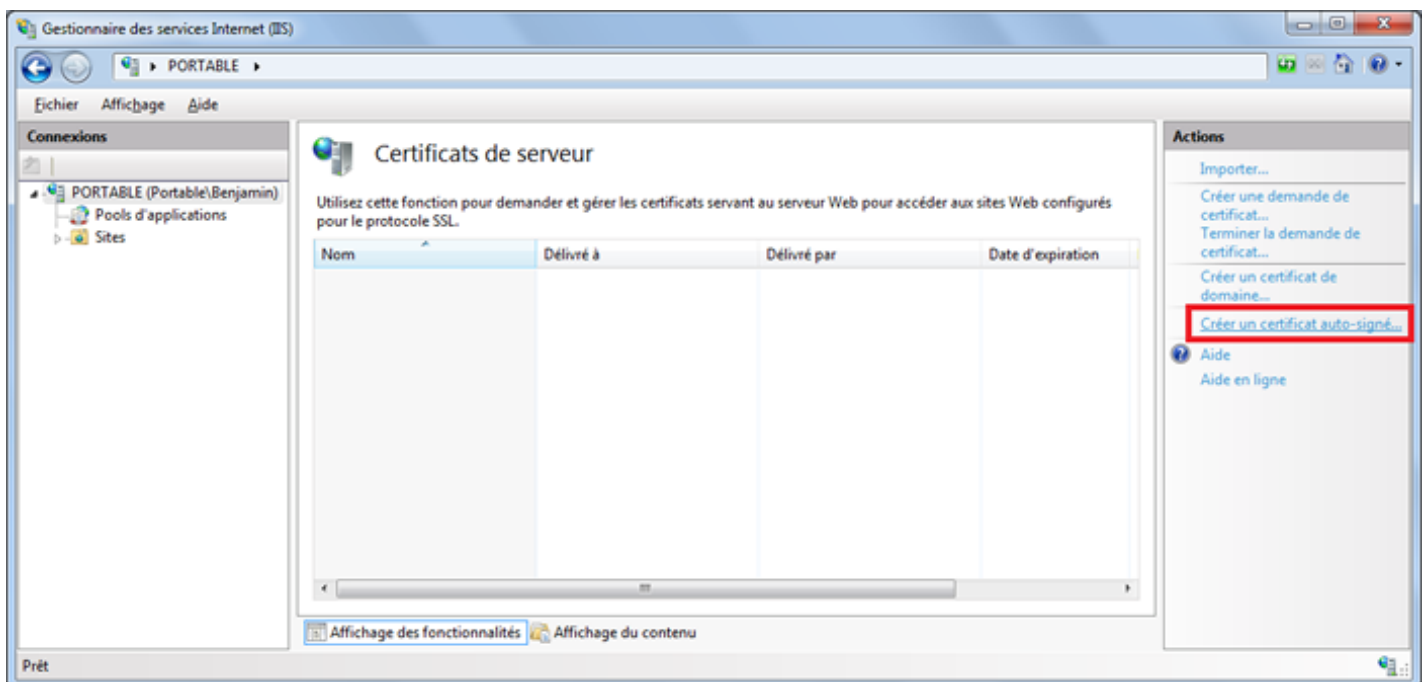
Pratique si par exemple vous ne souhaitez autoriser l'accès au dossier PHPMyAdmin qu'à votre IP en local (127.0.0.1 donc). De cette façon, deux filtrages peuvent être mis en place pour ce dossier : un au niveau de PHPMyAdmin via le fichier de configuration et un second au niveau de IIS.

VII - Mise en place d'un certificat SSL auto-généré pour votre site Web

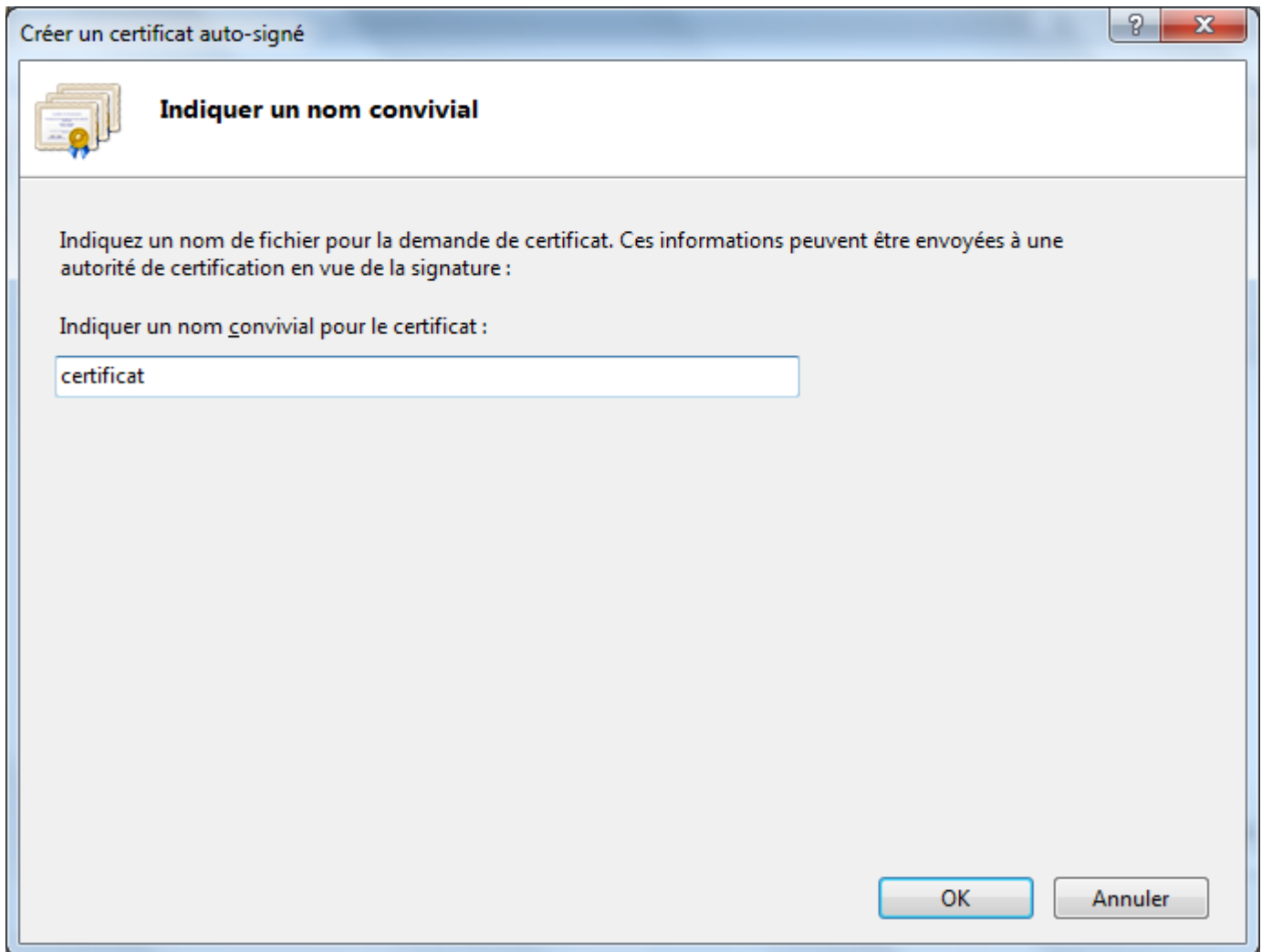
Sous IIS 7, vous pouvez également créer vos propres certificats SSL avec une simplicité déconcertante (aussi bien pour le protocole HTTP que FTP mais dans le cas présent seul le protocole HTTP nous occupe, la démarche étant quasiment identique pour le protocole FTP). Dans le cadre de notre serveur de test, nous allons mettre en place un certificat SSL de 128 bits auto-signé. Voici la procédure à suivre : dans la console de gestion d'IIS, à la racine de votre serveur, dans la colonne de gauche, vous devez avoir, au centre de la fenêtre, une icône intitulée **Certificats de serveur**. Double-cliquez sur cette dernière :



Dans la fenêtre nouvellement affichée, dans la colonne de droite, cliquez sur **Créer un certificat auto-signé**.



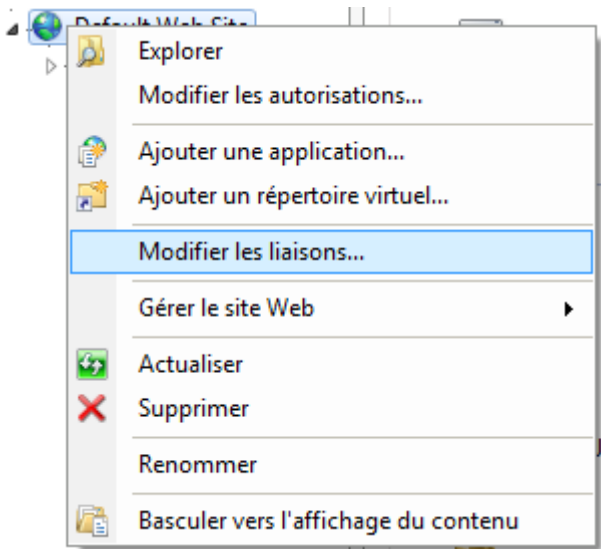
Vous allez maintenant devoir saisir un nom pour votre certificat :



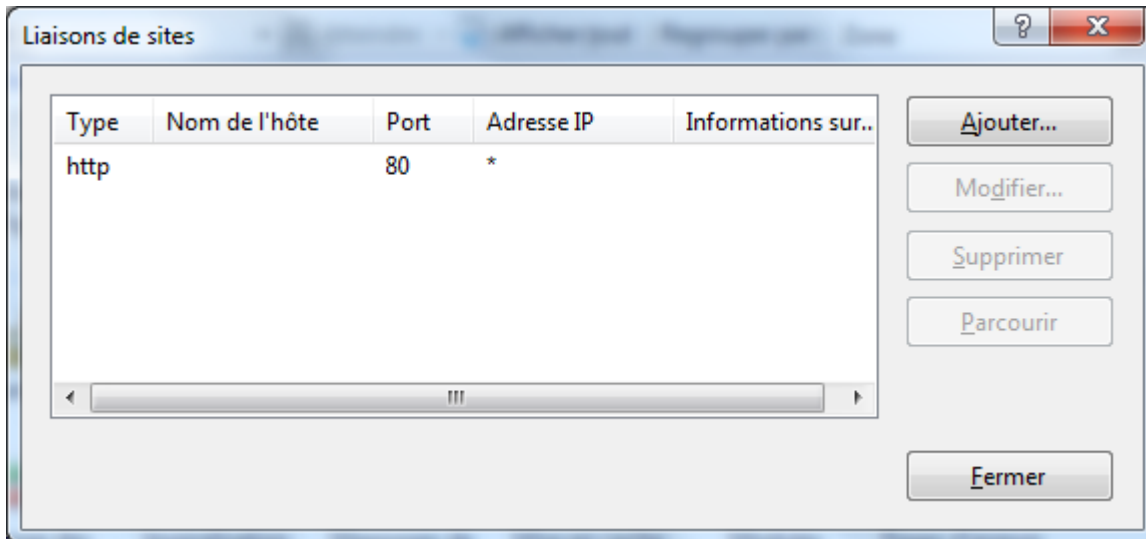
Remplissez le champ proposé (ici **certificat** m'a semblé suffisant, choisissez ce qui vous convient le mieux) et cliquez sur **OK**.

Et voilà, en un clic votre certificat est créé !

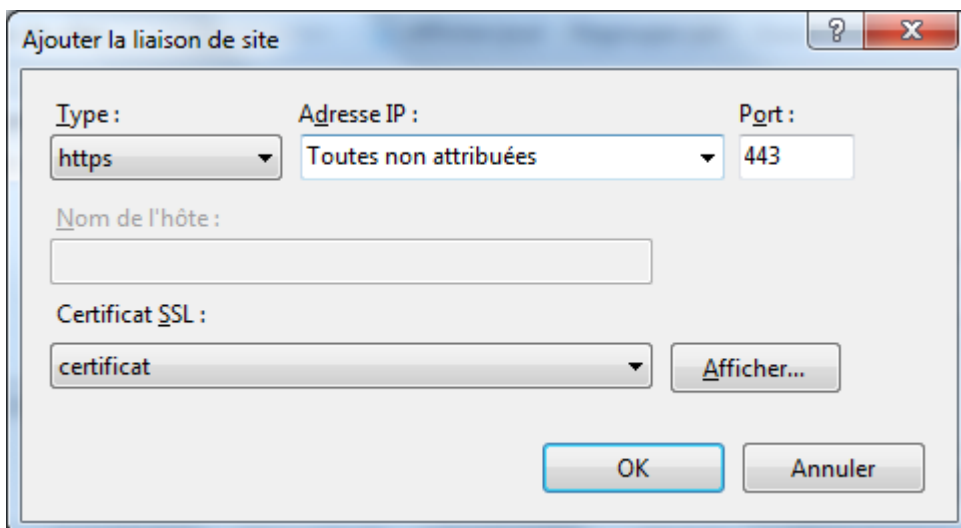
Il ne reste maintenant plus qu'à ajouter le support du SSL à votre site (le site par défaut en ce qui me concerne). Pour cela, faites un clic droit sur le site concerné et de cliquez sur **Modifier les liaisons** :



Dans la nouvelle fenêtre qui apparaît, cliquez sur **Ajouter...**

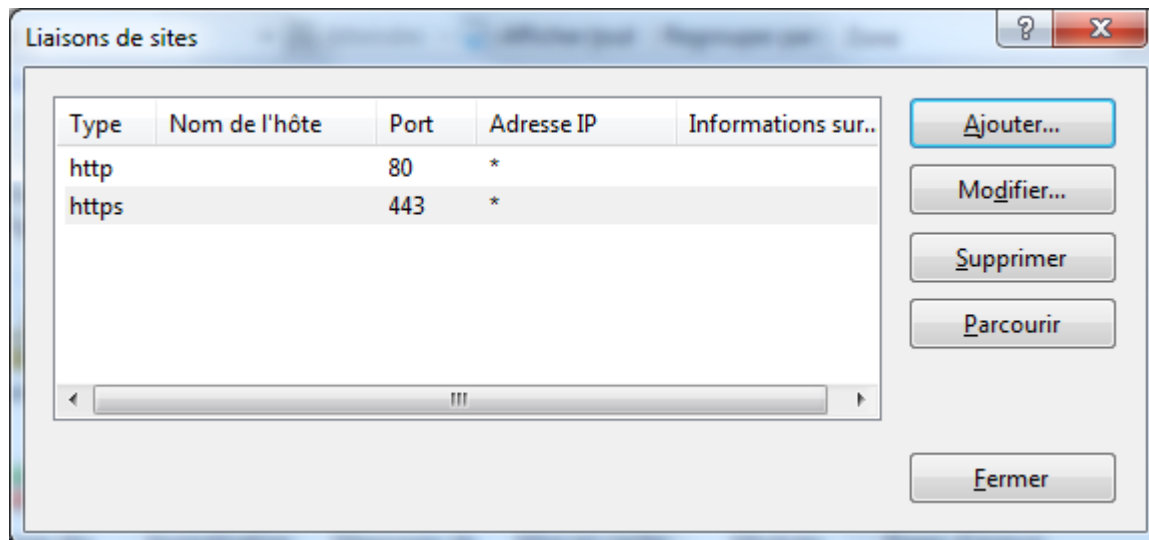


Et là vous remplissez les différents champs comme il se doit :

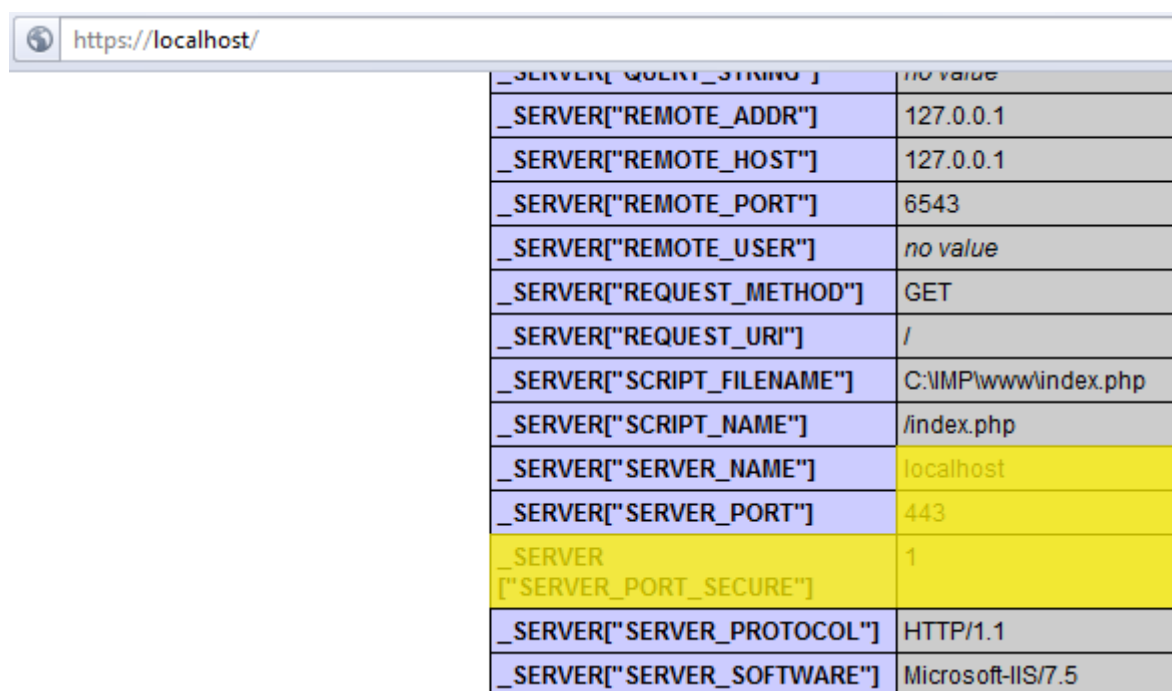


Vous devez alors sélectionner https (c'est le but du SSL), **Toutes non attribuées** pour l'adresse IP, **443** pour le port (port standard du protocole http sécurisé), et le nom du certificat auto-généré créé il y a peu. Cliquez sur **OK** une fois que vous avez saisi toutes les informations requises.

Vous devez maintenant avoir pour le même site deux protocoles et deux ports différents :



Vous pouvez maintenant accéder à votre site via **http://localhost** (protocole http non sécurisé, port 80) ainsi que par **https://localhost/** (protocole sécurisé, port 443) :



Le certificat utilisé est signé par le serveur qui l'utilise (signification d'auto-signé) et ne provient pas d'une autorité reconnue (comme Verisign par exemple) ce qui peut provoquer l'affichage d'un message de sécurité (je pense par exemple à Internet Explorer ou Mozilla Firefox). Rien de grave, il suffit de passer outre ce message et poursuivre la navigation.

VIII - Forcer l'utilisation du SSL avec PHPMyAdmin

Maintenant que vous disposez d'un certificat SSL pour votre site, vous pouvez, le plus simplement du monde, forcer l'utilisation du SSL lors de l'accès à PHPMyAdmin. C'est quelque chose d'important car des identifiants de connexion et des mots de passe circulent en clair.

Pour forcer l'utilisation du SSL dans PHPMyAdmin, il vous faut éditer le fichier de configuration **config.inc.php** qui est, pour moi en tout cas, dans **C:\IMP\www\phpmyadmin**.

Il vous faut alors ajouter la ligne suivante :

```
$cfg['ForceSSL'] = true;
```

Vous pouvez l'ajouter, par exemple, en dessous de

```
$cfg['SaveDir'] = '';
```

Enregistrez le fichier modifié et à partir de maintenant, dès que vous souhaitez accéder à PHPMyAdmin sans utiliser HTTPS (via <http://localhost/phpmyadmin>), vous serez automatiquement redirigé vers l'URL proposant une connexion sécurisée (<https://localhost/phpmyadmin>).

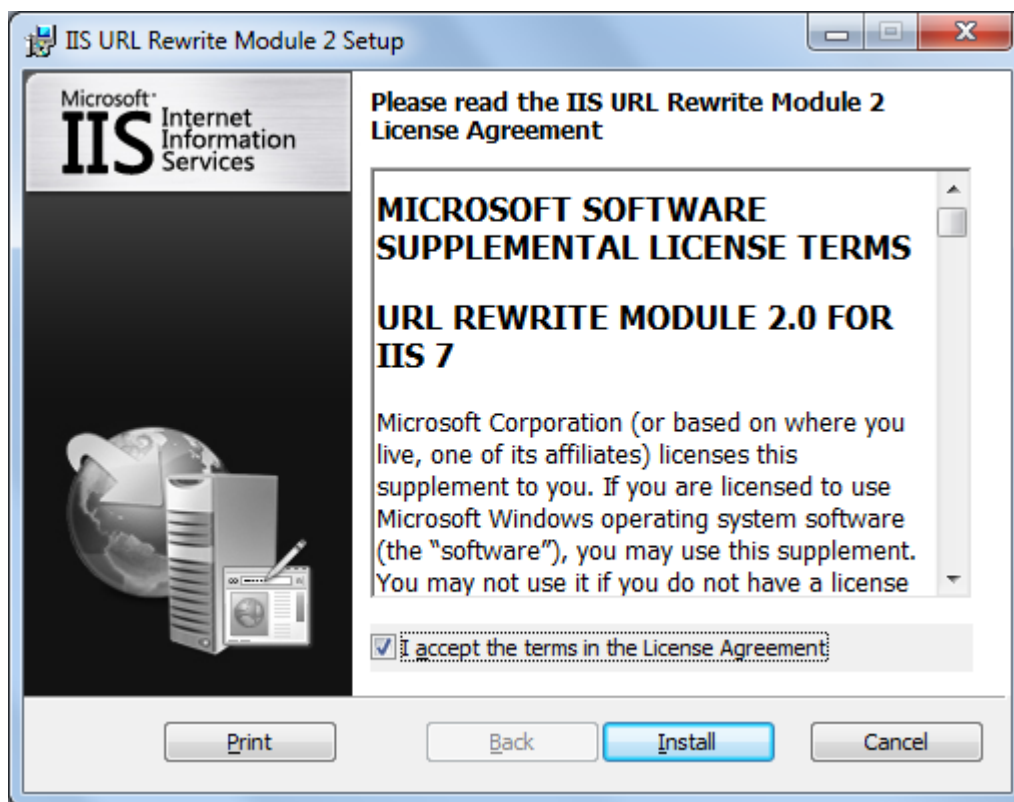
IX - IIS et la réécriture d'adresses (aka URL rewriting)

Je vous le disais en introduction, IIS gère maintenant très bien la réécriture d'adresse (vous avez sûrement déjà entendu ce concept sous sa forme anglaise : l'url rewriting).

Cette possibilité n'est pas offerte nativement. En effet, si vous souhaitez utiliser la réécriture d'adresse, il vous faudra télécharger un fichier MSI sur de Microsoft dédié à IIS. Voici l'adresse : <http://www.iis.net/download/URLRewrite>.

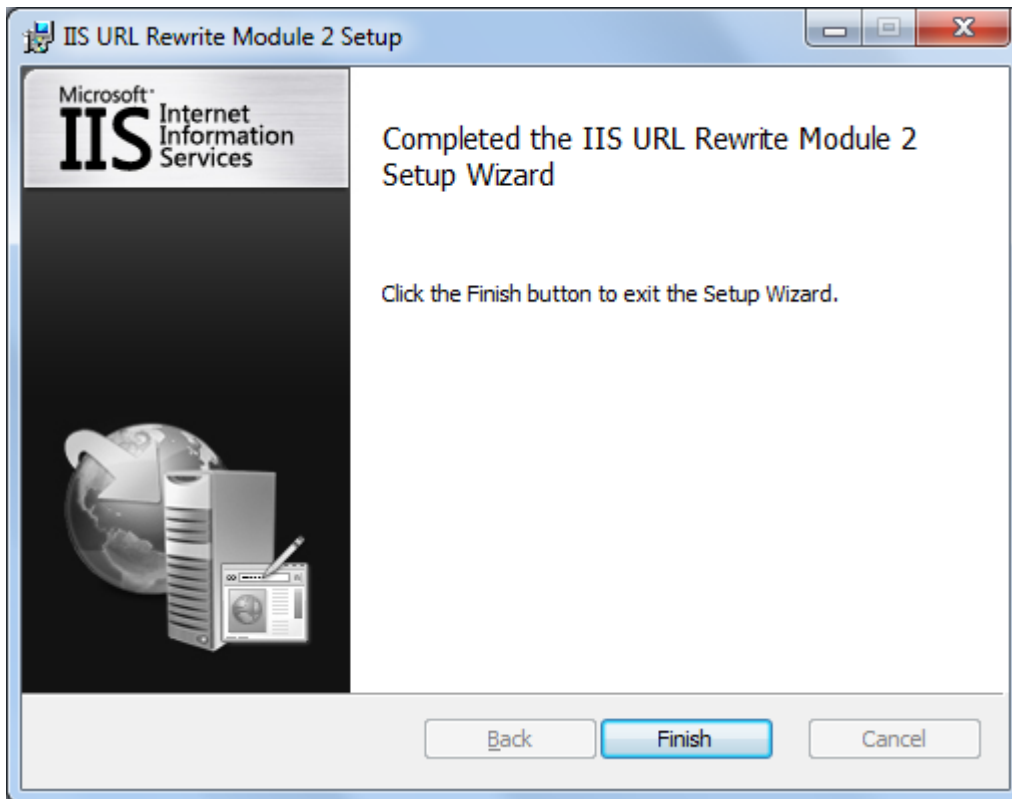
Dans la colonne de droite, vous trouverez deux liens, le premier, x86 pour les systèmes d'exploitation 32 bits et x64 pour les systèmes d'exploitation 64 bits. Je suis sous Windows 7 dans sa version 32 bits, je vais donc cliquer sur le lien x86. Un téléchargement vous sera alors proposé (moins de 6 Mo). Une fois le téléchargement effectué, nous allons installer ce module complémentaire (rien de bien compliqué, vous allez voir).

Double cliquez sur le fichier MSI pour lancer l'installateur :



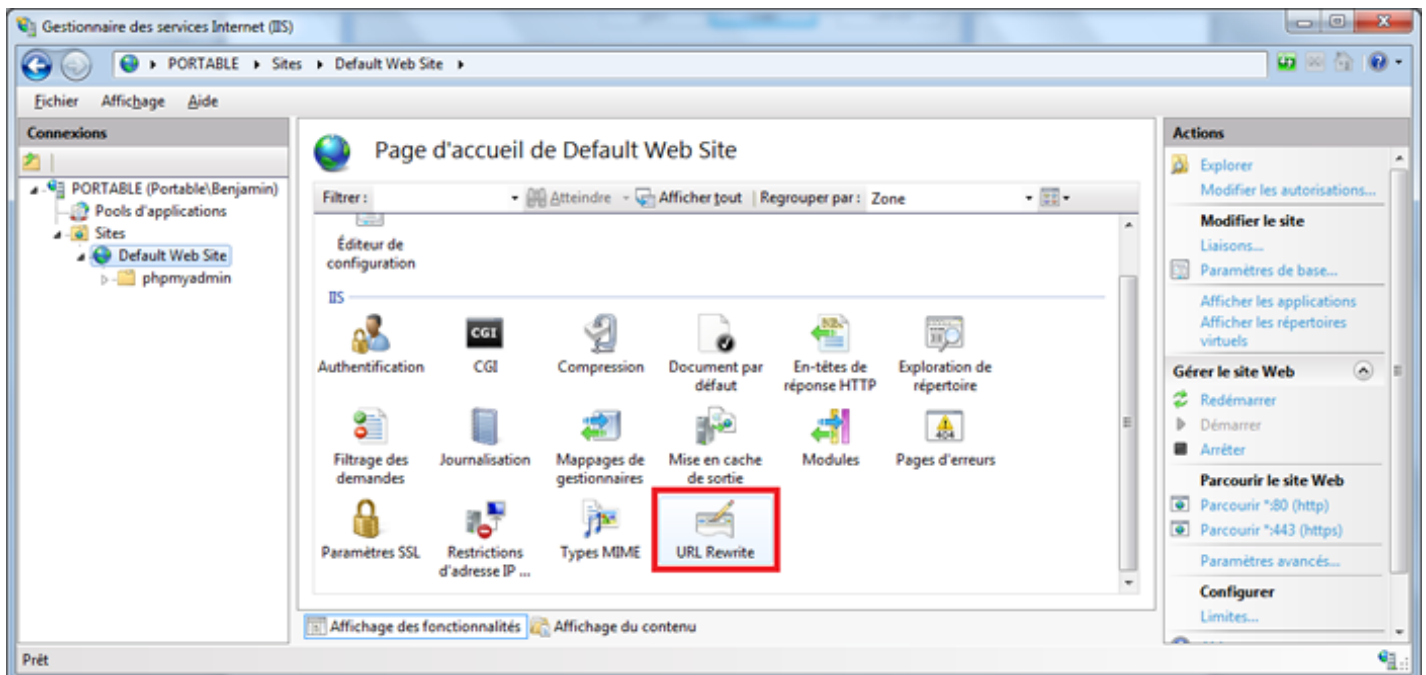
Cochez la case **I accept the terms in the License Agreement** (si, bien sûr, vous acceptez les termes donnés) puis cliquez sur **Install** pour débuter l'installation. L'UAC vous demandera alors confirmation pour débuter l'installation. Il vous faudra alors répondre **Oui**.

Cette étape passée, l'installation sera alors lancée et terminée en quelques instants :



Installation terminée, simple et rapide.

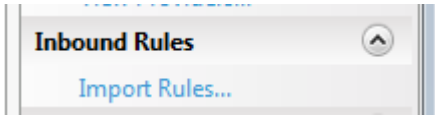
Si vous retournez maintenant dans la console de gestion d'IIS, dans le panneau de configuration de votre site, une nouvelle icône a du faire son apparition :



Double-cliquez dessus et cliquez ensuite, dans le menu de droite, sur **Add Rule(s)...** pour créer votre première réécriture d'adresse. Bien qu'en anglais, ce module vous offre un large choix de possibilités. C'est d'autant plus intéressant que bon nombre d'applications Web vous offrent la possibilité d'utiliser la réécriture d'adresses sous

IIS (Wordpress par exemple ; pour plus d'informations à ce sujet, consultez ce lien : http://codex.wordpress.org/Using_Permalinks).

Vous pouvez également (et c'est très intéressant), importer vos fichier .htaccess pour que le module sous IIS transforme vos règles de réécriture venant d'Apache au format IIS. Pour ce faire, dans le menu de droite du module de réécriture d'adresses, vous cliquez sur Import **Rules...** :



Les règles importées, vous pourrez alors les modifier, les adapter ou en rajouter selon votre convenance.

X - Lire les journaux de votre site Web

Il peut être intéressant (comme toujours) de regarder de temps à autre les fichiers de journalisation (fichiers log) de vos différents serveurs (et plus particulièrement ceux de votre serveur Web).

Concernant votre site sous IIS, il suffit de vous rendre dans la catégorie **Journalisation** de votre site pour consulter et/ou modifier les paramètres. Par défaut, un nouveau journal sera créé chaque jour. Plutôt pratique pour la rotation des fichiers. Seulement, si vous en ouvrez un, vous vous rendrez vite compte que ces fichiers ne sont pas très pratiques pour le traitement de données (selon moi en tout cas).

Une réponse à ce problème ? Oui et elle s'appelle Log Parser. Vous trouverez cet utilitaire sur le site de Microsoft (ou directement à cette adresse : <http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=24659>).

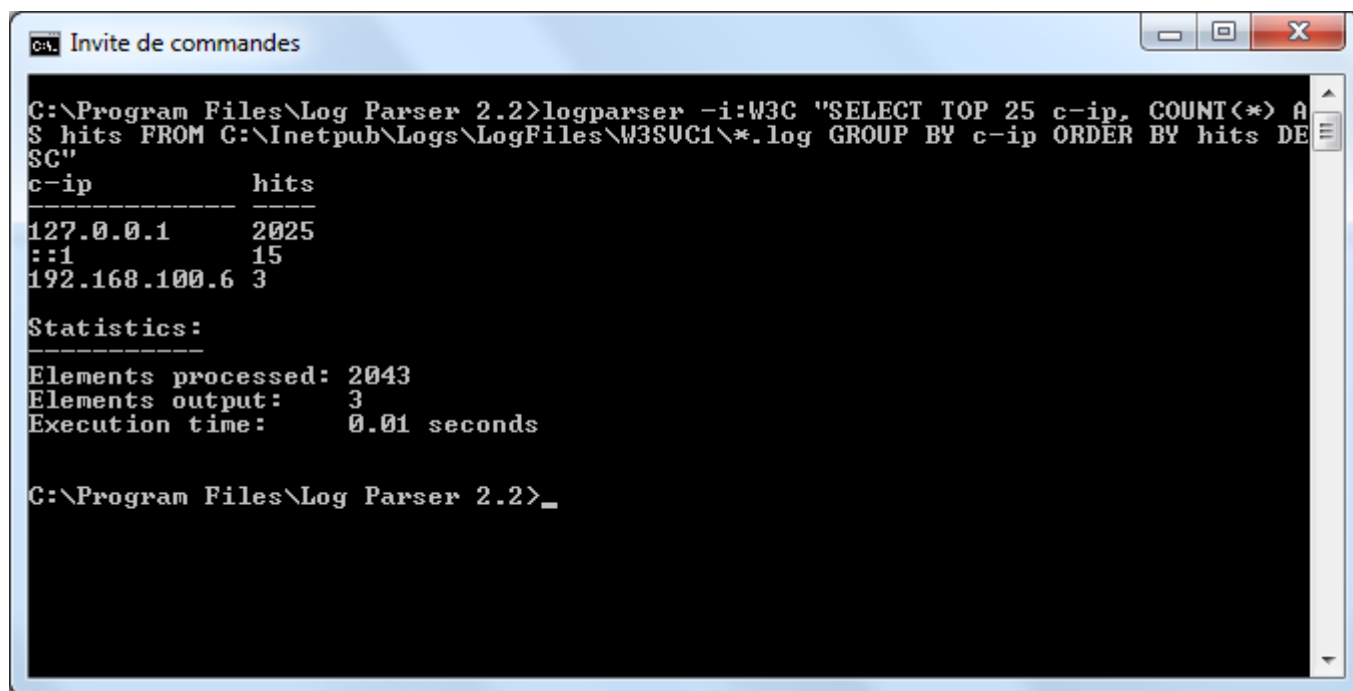
C'est un package au format MSI qu'il vous faudra installer (installation complète pour ce qui me concerne). Je ne détaille pas plus l'installation, quelques clics suffisent même si c'est en anglais pour mener à bien cette installation.

Là où cet utilitaire devient intéressant, c'est dans la manière de l'utiliser. En effet, pour pouvoir disposer de statistiques (par exemple) intéressantes (dans le sens où elles sont exploitables simplement), il vous suffit de lui passer comme argument une requête au format SQL. Si vous lisez ce tutoriel, je doute que les requêtes SQL vous soit inconnues.

Si vous êtes familier avec le langage SQL vous pouvez créer des statistiques plutôt sympathiques. Pour vous donner un exemple (tout simple), nous allons créer une requête qui va nous donner la liste des vingt-cinq adresses IP qui ont fait le plus de hits sur nos sites (un seul dans mon cas mais ça ne change rien à la requête vu que je passe en revue tous les fichiers de logs) :

```
SELECT TOP 25 c-ip, COUNT(*) AS hits
FROM C:\inetpub\logs\LogFiles\W3SVC1\*.log
GROUP BY c-ip
ORDER BY hits DESC
```

Je choisis donc les 25 adresses (au maximum) et leurs nombres de vues à partir de tous les fichiers logs dans C:\inetpub\logs\LogFiles\W3SVC1*.log (répertoire par défaut des fichiers log d'IIS). Je groupe ensuite par IP puis j'ordonne la liste par ordre décroissant du nombre de vue par IP. Si vous saisissez cette requête dans une invite de commande et vous aurez quelque chose de semblable à ceci (suivant les logs disponibles et les visites de votre site) :



```

C:\Program Files\Log Parser 2.2>logparser -i:W3C "SELECT TOP 25 c-ip, COUNT(*) AS
 hits FROM C:\inetpub\Logs\LogFiles\W3SVC1\*.log GROUP BY c-ip ORDER BY hits DE
SC"
c-ip                hits
-----
127.0.0.1           2025
::1                 15
192.168.100.6      3

Statistics:
-----
Elements processed: 2043
Elements output:    3
Execution time:     0.01 seconds

C:\Program Files\Log Parser 2.2>_


```

Dans mon cas, trois adresses IP différentes ont consulté mon site par défaut (trois adresses locales, je vous le concède, serveur de test local oblige) mais ces statistiques peuvent vous rendre bien des services.

Pour aller plus loin, vous pouvez bien sûr créer vos requêtes, les stocker dans un fichier bat et appeler ce fichier bat par une tâche planifiée sur votre serveur pour avoir des données actualisées de manière régulière.

Là où LogParser peut être encore plus intéressant, c'est qu'il peut, à partir de vos fichiers log toujours, créer des images (de différents formats). Quoi de mieux que des graphiques pour générer des statistiques ?

Seulement, si vous essayez de créer des graphiques, vous allez vous heurter à un message d'erreur (c'est mon cas avec Windows 7 et Microsoft Office 2010) : **Error creating output format "chart": This output format requires a licensed Microsoft Office Chart Web Component to be installed on the local machine.** D'après mes différentes lectures, il s'avère que la création des graphiques repose sur les composants Web de Microsoft Office 2003 (oui, ça ne rajeunit personne). Avec Office 2010, c'est bien normal que cela ne fonctionne pas. Cependant, il est toujours possible de télécharger ce plug-in afin de générer des graphiques. Vous trouverez ce plug-in en téléchargement à cette adresse : <http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=22276>.

 *Il est à noter que l'installation de ce composant additionnel doit être suivie d'une mise à jour de votre système via Windows Update. En effet, ce plug-in étant assez ancien, des mises à jour importantes sont à installer.*

Maintenant, pour reprendre mon exemple donné plus haut, si je souhaite créer un graphique qui me donne une idée du nombre de visites par navigateur (<http://logparserplus.com/Examples/42> pour le cas présent, à mettre à jour pour avoir une idée plus précise) :

```

logparser -i:w3c -o:chart -chartType:barstacked "SELECT
CASE strcnt(cs(user-agent),'Firefox') WHEN 1 THEN 'Firefox'
ELSE CASE strcnt(cs(user-agent),'netscape') WHEN 1 THEN 'netscape'
ELSE CASE strcnt(cs(user-agent),'AOL') WHEN 1 THEN 'AOL'
ELSE CASE strcnt(cs(user-agent),'Opera') WHEN 1 THEN 'Opera'
ELSE CASE strcnt(cs(user-agent),'Chrome') WHEN 1 THEN 'Chrome'
ELSE CASE strcnt(cs(user-agent),'Mobile') WHEN 1 THEN 'SmartPhone'
ELSE CASE strcnt(cs(user-agent),'Safari') WHEN 1 THEN 'Safari'
ELSE CASE strcnt(cs(user-agent),'MSIE+5') WHEN 1 THEN 'IE 5'
ELSE CASE strcnt(cs(user-agent),'MSIE+6') WHEN 1 THEN 'IE 6'

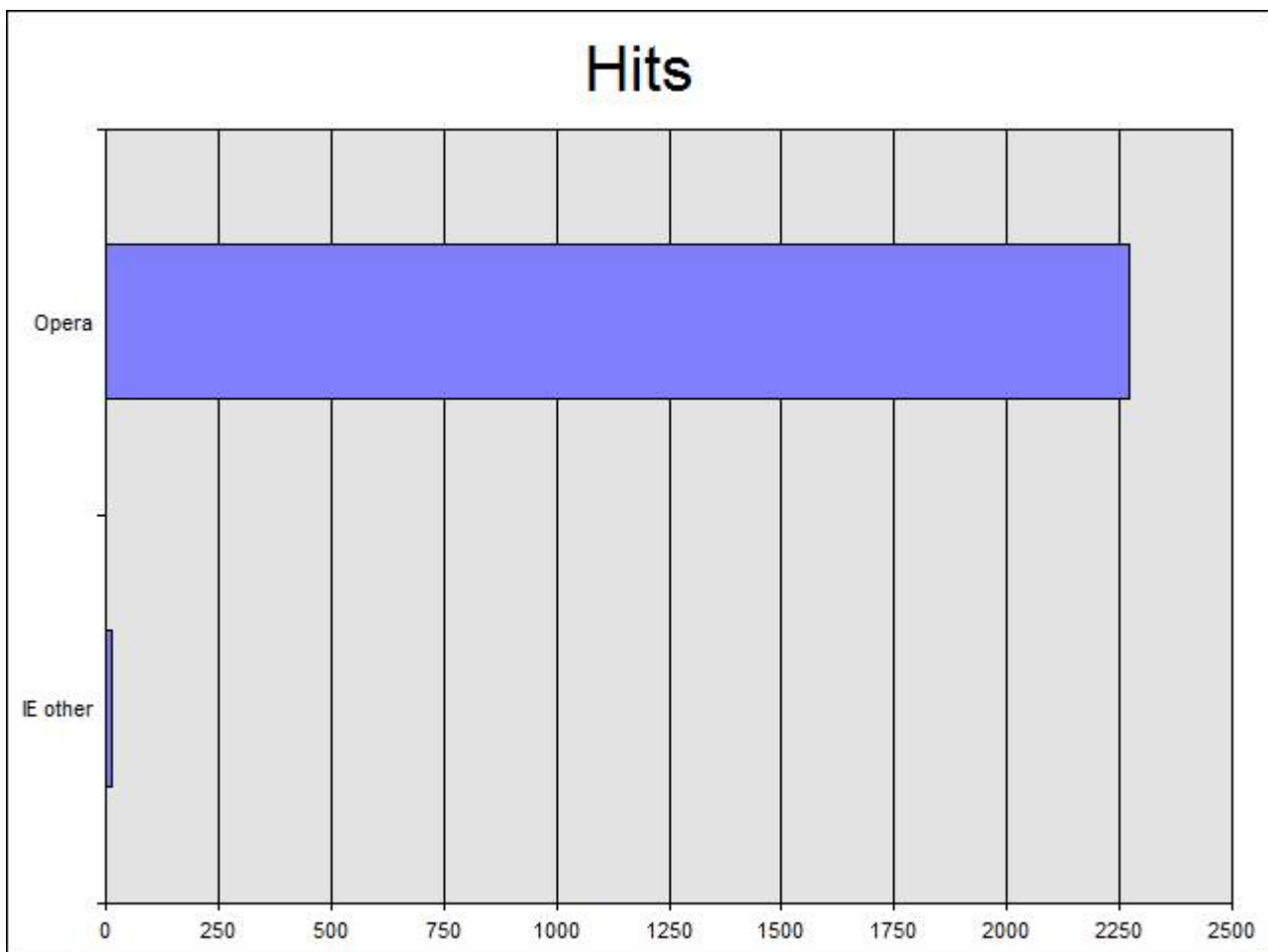
```

```

ELSE CASE strcnt(cs(user-agent),'MSIE+7') WHEN 1 THEN 'IE 7'
ELSE CASE strcnt(cs(user-agent),'MSIE+8') WHEN 1 THEN 'IE 8'
ELSE CASE strcnt(cs(user-agent),'MSIE') WHEN 1 THEN 'IE other'
ELSE CASE strcnt(cs(user-agent),'bot') WHEN 1 THEN 'Bot'
ELSE CASE strcnt(cs(user-agent),'spider') WHEN 1 THEN 'spider'
ELSE CASE strcnt(cs(user-agent),'PutHTTP') WHEN 1 THEN 'PutHTTP'
ELSE CASE strcnt(cs(user-agent),'Mozilla/4.0') WHEN 1 THEN 'Mozilla/4.0 other'
ELSE 'Unknown'
End
End
End
End
End
End
End
End
End
End
End
End
End
End
End
End
End as Browser, COUNT(cs(User-Agent)) as Hits INTO Browsers.jpg
FROM C:\inetpub\logs\logfiles\W3SVC1\*.log
GROUP BY Browser ORDER BY Hits ASC

```

Pour le résultat suivant (image stockée dans mon cas dans le répertoire suivant : C:\Users\Benjamin\AppData\Local\VirtualStore\Program Files\Log Parser 2.2) :



Bien sûr, d'autres types de graphiques sont disponibles et vous pouvez générer des statistiques avec beaucoup d'autres ressources autres que la ligne de commande DOS (C++, C#, VBScript, etc.).

Si vous souhaitez avoir une multitude d'exemples et de réalisations possibles avec LogParser, je vous laisse chercher avec votre moteur de recherche favori, les exemples sont nombreux (le billet de Mike Lichtenberg disponible à cette adresse <http://mlichtenberg.wordpress.com/2011/02/03/log-parser-rocks-more-than-50-examples/> vous montre quelques réalisations assez étonnantes ainsi que <http://logparserplus.com/Examples>).

J'avoue que ce n'est pas ce qu'il y a de plus intéressant pour un serveur de test en local (c'est pour cette raison que je ne m'étends pas sur le sujet) mais je pense sincèrement qu'un outil pareil doit être démocratisé le plus possible (à ne pas manquer sur un serveur dédié Windows).

XI - Conclusion

Vous disposez maintenant d'un serveur de test local avec une configuration que l'on peut qualifier "de base". Vous pouvez, bien sûr, modifier les configurations des différents éléments présentés ci-dessus comme vous le souhaitez. Les besoins de chacun étant tous différents.

Vous pouvez consulter et télécharger d'autres modules pour IIS 7.5 sur le site dédié à IIS à l'adresse suivante : <http://www.iis.net/download> (vérifiez que les modules sont bien disponibles pour IIS 7.5, ce n'est pas le cas de tous).

J'espère que vous aurez pris autant de plaisir à lire ce tutoriel que j'en ai pris pour l'écrire. Si vous rencontrez la moindre difficulté, une erreur ou une modification à apporter, n'hésitez pas à me contacter via le forum.