



## Codes Postaux Français

## Démonstration Silverlight 2.0

### Code source

Version 1.0 – Janvier/Août 2009 © Olivier Dahan / [www.e-naxos.com](http://www.e-naxos.com)

### Présentation

« Les codes postaux français » est une application exemple écrite pour Silverlight 2.0 (ou 3.0).

Elle met en œuvre de nombreuses techniques de la 2D ou la 3D animée pour l'interface jusqu'aux services Web avec base de données.

### L'exemple Live

Vous pouvez voir l'application fonctionner à cette adresse :

<http://www.e-naxos.com/svcp/SVCPTestPage.html>

### Avertissements

Le code source est publié « tel quel » sans aucune garantie ni aucun support uniquement à titre d'exemple.

Le code est conçu pour fonctionner avec un service se trouvant sur les serveurs de [www.e-naxos.com](http://www.e-naxos.com), nous vous demandons lors de vos tests, si vous compilez le code, de bien vouloir modifier ce code afin qu'il accède à la copie locale du Web service sur votre machine afin d'éviter la surcharge de notre serveur. S'agissant d'un exemple une très faible bande passante a été attribuée à l'application et toute utilisation intensive nous obligerait à couper le service. Nous vous remercions de votre compréhension.

### Organisation

Le code se présente sous la forme de deux applications :

- Un service Web lié à une base de données de type Access / Jet hébergé dans une application de type ASP.NET
- Un frontal Silverlight 2 (ou plus).

WSCPNET est le projet ASP.NET, il contient deux sous-projets : WSCPNET, le service Web proprement, et WSCPClient, une petite application de test du service Web utilisée pour la mise au point de l'application en mode WPF. Cette application de test ne présente pas un grand intérêt sauf à voir comment utiliser un Web Service depuis une application WPF, raison pour laquelle elle a été conservée dans le paquet zip fourni.

La base de données est une version très allégée de la base originale pour éviter de grossir la taille du Zip du code source. Seul le département de l'Ain et ses villes sont dans la base exemple. Ne cherchez pas Marseilles ou Brest, si ce n'est pas dans l'Ain ce n'est pas dans la base de test ! Pourquoi l'Ain ? Parce que c'est le premier enregistrement de la base, tout le reste a été supprimé ☺

## Q&A

### Pourquoi avoir choisi une base de données Access/Jet ?

Cela permet d'avoir un fichier disque simple ne nécessitant aucune installation ce qui est parfait pour une application de test. De plus, si vous souhaitez utiliser le code sur un serveur distant il faut savoir que les hébergeurs font généralement payer un supplément pour les bases SQL Server alors que l'utilisation d'Access/Jet est gratuite... S'il n'est pas question de conseiller une telle base de données pour une mise en production, comme vous pouvez le constater en utilisant notre exemple en ligne, cela fonctionne parfaitement bien.

### Comment exécuter l'exemple ?

Le code fourni l'est seulement à titre éducatif, pour voir l'exemple fonctionner utiliser l'application en live que nous avons mise en ligne.

Maintenant, le code fourni est complet et peut être compilé. Les références Web pointent vers le service Web se trouvant sur nos serveurs, pour ne pas perturber ces derniers modifiez le code pour faire pointer les références vers une copie locale sur votre machine.

### Quels outils pour voir / modifier le code ?

Le code proprement dit peut être ouvert et modifié avec Visual Studio 2008 équipé de ses Service Packs et de l'ensemble des outils annexes pour Silverlight.

Pour le visuel de l'application Silverlight vous aurez besoin de Expression Blend.

### Pourquoi un service Web ASMX au lieu de WCF ?

D'abord cette application n'est qu'un exemple, pas un produit final. Les services ASMX sont d'une grande simplicité de mise en œuvre et se prêtent donc bien à un tel exercice. Ensuite il est évident que dans la réalité, une application Silverlight peut être amenée à dialoguer avec tout type de Web Service, pas forcément écrits en .NET ni même utilisant WCF. L'utilisation de ASMX se rapproche plus de ce cas réel.

## Manque-t-il des ressources dans le paquet fourni ?

Certains fichiers ne sont pas fournis en effet :

WSCPNet.pfx et enaxosSVCP.pfx qui sont les clés de signature des deux projets sont absentes, pour signer l'application si vous la compilez, générez vos propres clés.

FlawlessCode.Navigation.dll est une bibliothèque de code gratuite utilisée dans l'application Silverlight pour gérer les transitions entre les pages. Vous pouvez vous procurer le code source et le compiler à cet endroit : <http://www.flawlesscode.com/post/2008/03/Silverlight-2-Navigating-Between-Xaml-Pages.aspx>

## A quoi sert le fichier crossdomain.xml ?

Silverlight 2 (tout comme Flash) interdit par défaut les appels cross domain pour des raisons de sécurité. Le service Web doit ainsi, à la racine du site l'hébergeant, proposer un fichier XML décrivant quelle est la stratégie de sécurité, notamment quels domaines ont le droit d'accéder au service Web publié.

Si Silverlight ne trouve pas ce fichier sur le serveur, les appels au service Web échouent.

Nous utilisons ici la version la plus permissive puisque CrossDomain.XML est le fichier utilisé par ADOBE pour Flash qui est reconnu par Silverlight. Cela permet d'envisager la conception d'un frontal identique en Flash sans modification sur le serveur.

Pour respecter les indications de Microsoft, il faut en réalité fournir un autre fichier CrossDomainPolicy.XML qui est le premier recherché par une application Silverlight. Le fichier au format Adobe n'est lu que si le fichier au format Microsoft n'est pas présent. En production nous vous conseillons ainsi de placer les deux fichiers sur le serveur. Le fichier Microsoft étant suffisant si vous ne souhaitez pas que des frontaux Flash puissent accéder à vos Web Services.

Pour plus d'information sur le format du fichier CrossDomainPolicy.XML vous pouvez lire ce post du blog de Pascal Cabanel : <http://pcabanel.over-blog.com/article-19896147.html> ou bien vous reportez à la documentation officielle Microsoft si vous désirez entrer dans les détails.

## Comment ont été créés les divers graphiques ?

L'image de l'enveloppe est un scan d'une vraie enveloppe traité sous PhotoShop notamment pour ajouter l'ombrage (effet présent dans Silverlight 3 mais absent de SL 2).

Les timbres ont été conçus sous Expression Design.

Le timbre avec l'oiseau animé a été créé en passant le timbre précédent sous Swift 3D (un excellent petit logiciel de 3D qui sait aussi générer du XAML) au sein duquel l'oiseau a été ajouté. Ici c'est une simple vidéo qui a été produite et réintégrée à l'application (après encodage avec Media encoder).

Les petits symboles 3D animés (aide, home) ont été créés sous Swift 3D et exportés en vidéo Avi puis encodés avec Expression Media Encoder.

Les graphiques du type barcode, étiquettes, bout de scotch, oblitération, post-it, éclaboussures, etc, ont été créés sous Illustrator ou Expression Design puis exportés en PNG.

## Y-a-t-il des explications sur le code source ?

Actuellement non. Rien d'autre que le code source et ses commentaires. Un article est prévu autour de cet exemple mais le temps pour l'écrire étant difficile à trouver, le choix a été pris de publier le source pour donner l'occasion à chacun de l'étudier.

## Le code est-il parfait ?

En de nombreuses années d'Audit auprès de grands éditeurs pourtant fort réputés nous n'avons jamais vu de « code parfait » quel que soient les moyens et le sérieux déployés. Tout est toujours affaire de compromis entre l'intention de départ et le temps imparti pour le réaliser. Nous n'échappons pas à la règle surtout s'agissant d'un petit exemple vite développé !

La décision de publier le code source a été prise récemment et aucun « ménage » ni habillage cosmétique n'a été fait dans le but de supprimer les quelques erreurs ou tâtonnements qui peuvent éventuellement l'émailler.

Exemple visuel donc, exemple de code, pas forcément. Toutefois il n'y a pas non plus à en rougir ☺

## Réutilisation ?

Si ce petit exemple ne mérite pas de déposer un brevet, aucune utilisation ou diffusion ne peut en être faite sans l'autorisation de l'auteur, bien entendu. Il en va de même des graphiques fournis dans le cadre de l'exemple.

Maintenant, les techniques démontrées sont parfaitement réadaptables à d'autres cas particuliers, ne vous gênez pas pour le faire !

## Plus d'exemples ?

Il suffit de suivre Dot.Blog régulièrement ou d'y chercher ce qui a été déjà publié, des exemples Silverlight, il y en a plein !

Dot.Blog : [www.e-naxos.com/blog](http://www.e-naxos.com/blog)