Utilisation d'une dll dans un ActiveX

Auteur : Philippe Lacomme et Raksmey Phan.

But : Ce tutorial a pour but de vous aider à intégrer dans un projet ActiveX existant une dll créée dans un autre projet.

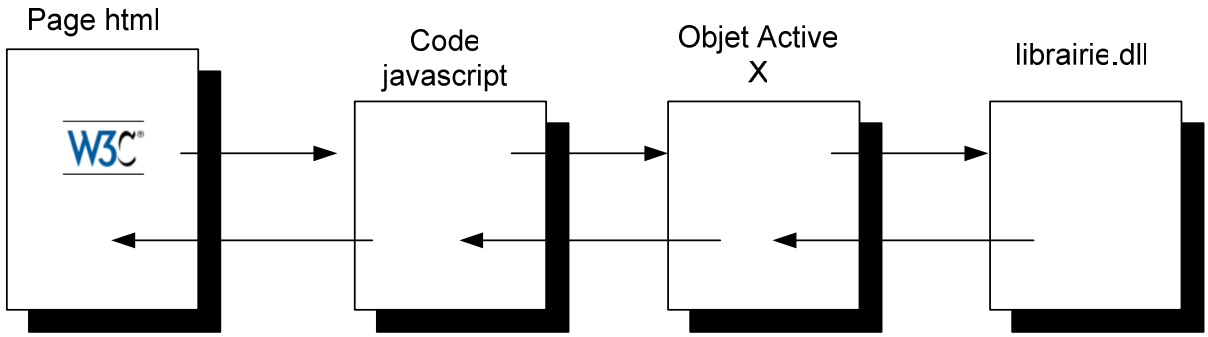
Environnement : Visual Studio.

PRINCIPE GENERAL

Le contenu de la page html est généré par un code javascript.

Le code javascript utilise utilise un object activex nommé **Object_Active_X**

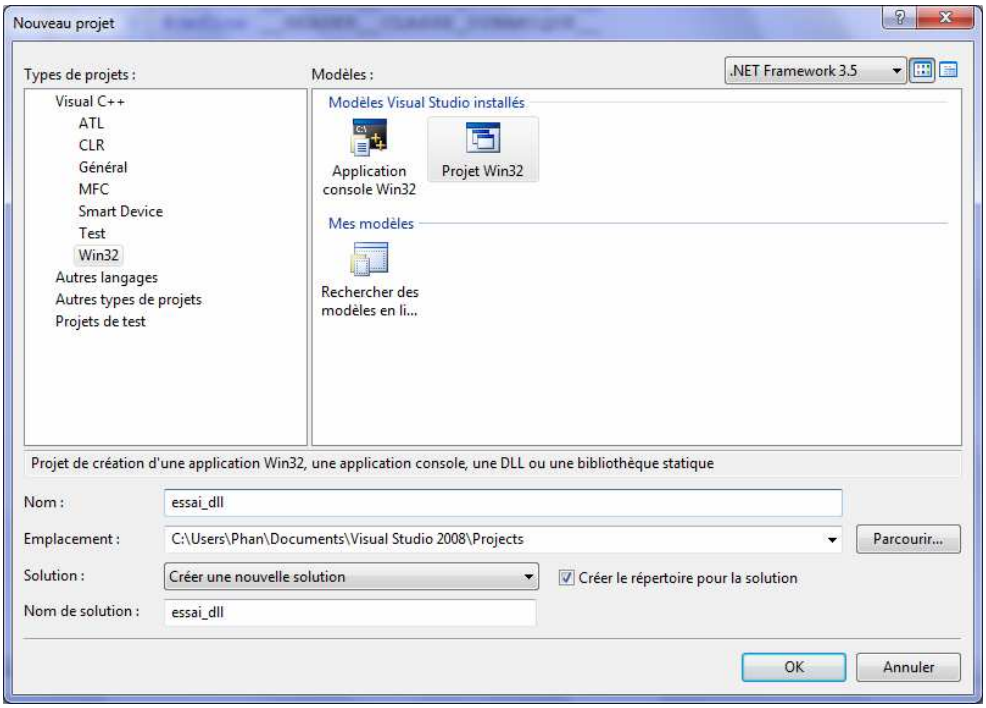
L'object activeX utilise une librairie dll nommée : **essaidll.dll**

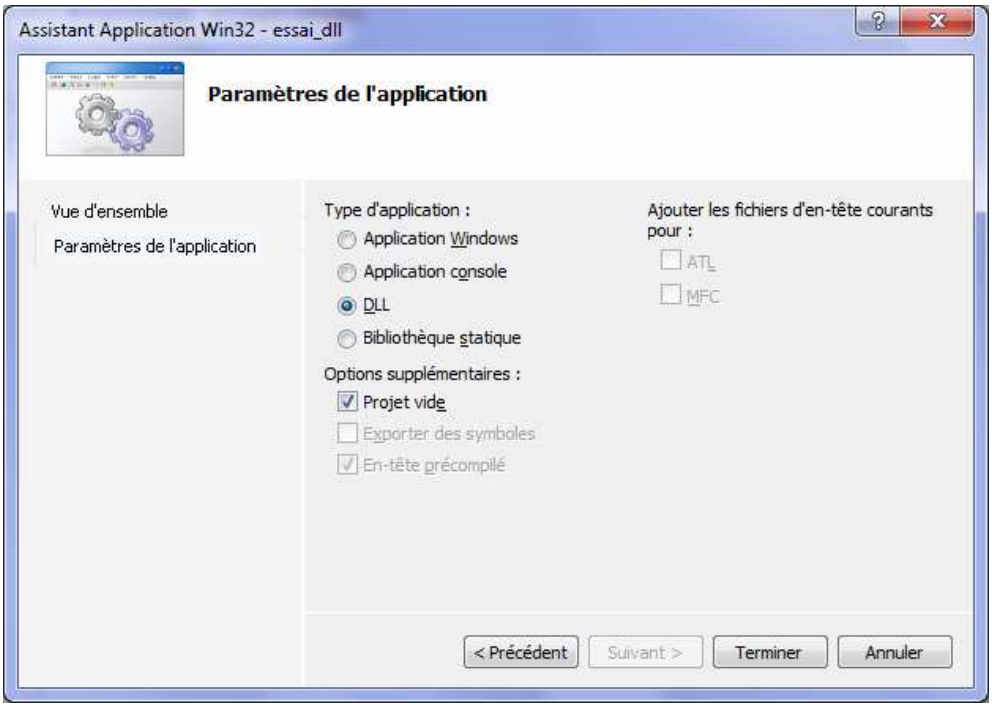


CREATION D'UNE DLL

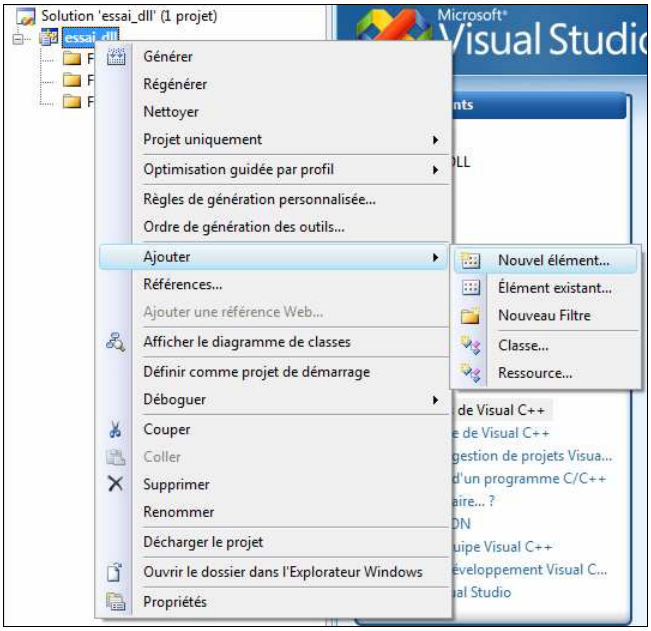
Cette partie du tutoriel s'inspire de l'excellent tutoriel sur le site « cours.polymtl », téléchargeable à deux adresses :
http://www.cours.polymtl.ca/log2410/docs/documents/TutorielDLL.pdf
http://fc.isima.fr/~phan/activex/TutorielDLL.pdf

Dans un premier temps, il faut créer le dll qui nous permet de faire nos calculs.

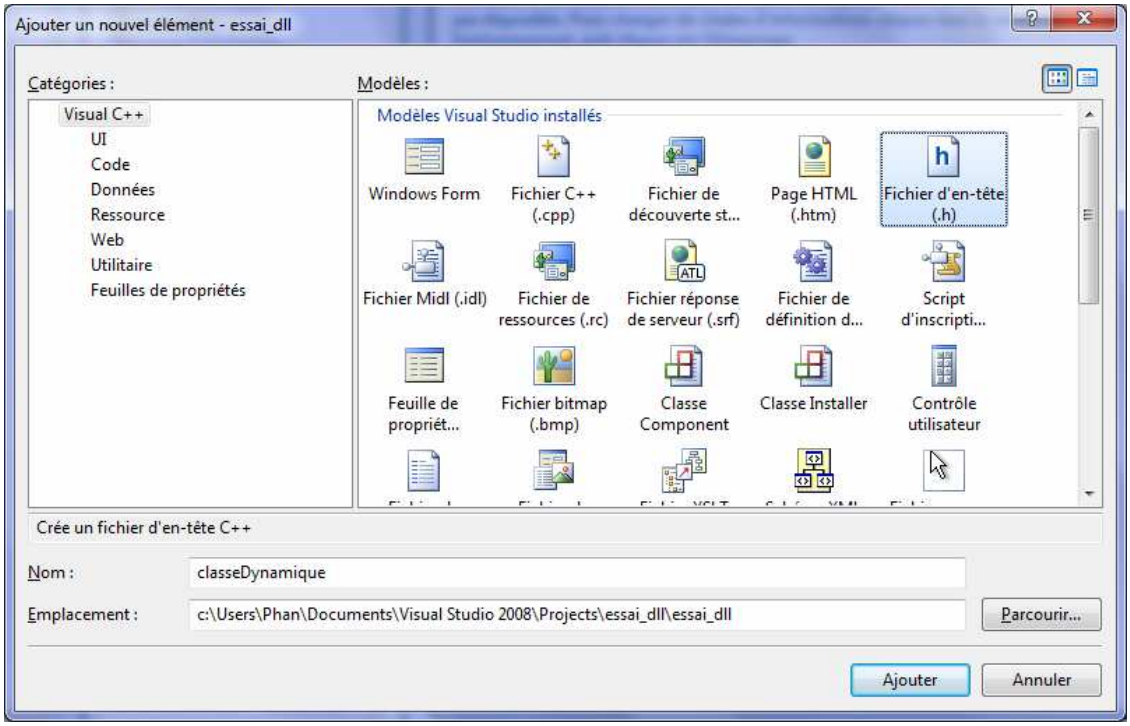




Faire un clic droit sur essai_dll et choisir ajouter / nouvel élément.



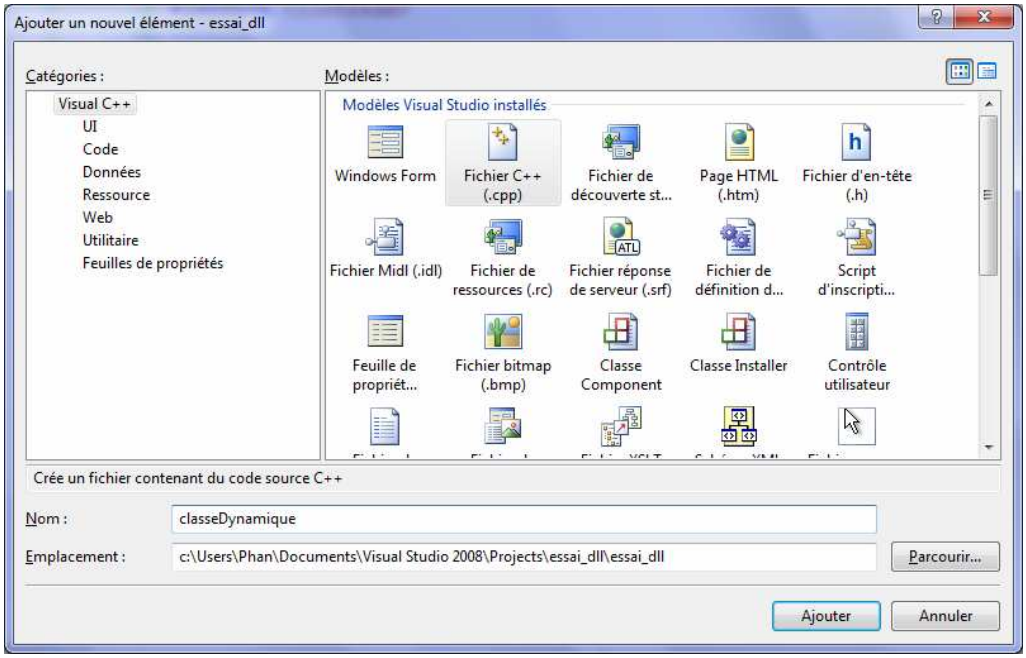
On ajoute un fichier **classeDynamique.h**



On crée deux fonctions calcul, une à l'intérieure d'une classe et l'autre à l'extérieur. Cela pour montrer que l'utilisation de la dll peut passer par une classe ou bien directement par une fonction.

```
// ClasseDynamique.h
#ifndef __HEADER__CLASSE_DYNAMIQUE__
#define __HEADER__CLASSE_DYNAMIQUE__
#include <string>
#include <iostream>

class ClasseDynamique
{
public:
   __declspec(dllexport) ClasseDynamique(void);
   __declspec(dllexport) ~ClasseDynamique( void );
   __declspec(dllexport) int Addition_Interne(int, int) ;
private:
};

__declspec(dllexport) int Soustraction_Externe ( int, int );

#endif
```

On crée le .cpp.

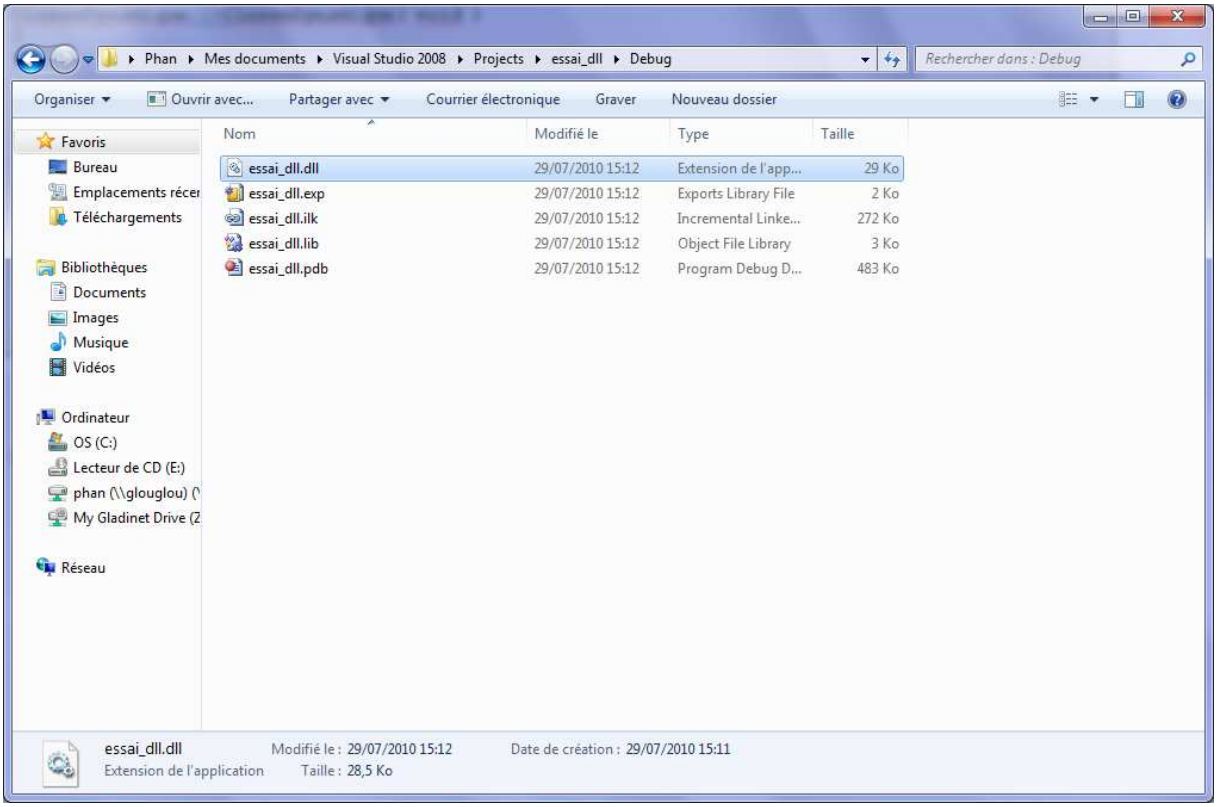


Le code est le suivant :

```
#include "ClasseDynamique.h"

ClasseDynamique::ClasseDynamique(void)
{
}

ClasseDynamique::~ClasseDynamique( void )
{
}

int ClasseDynamique::Addition_Interne(int INa, int INb)
{
   int value = INa + INb ;
   return value ;
}

int Soustraction_Externe(int INa, int INb)
{
   int value = INa - INb ;
   return value ;
}
```

Maintenant que le dll est créé, on peut compiler le projet.

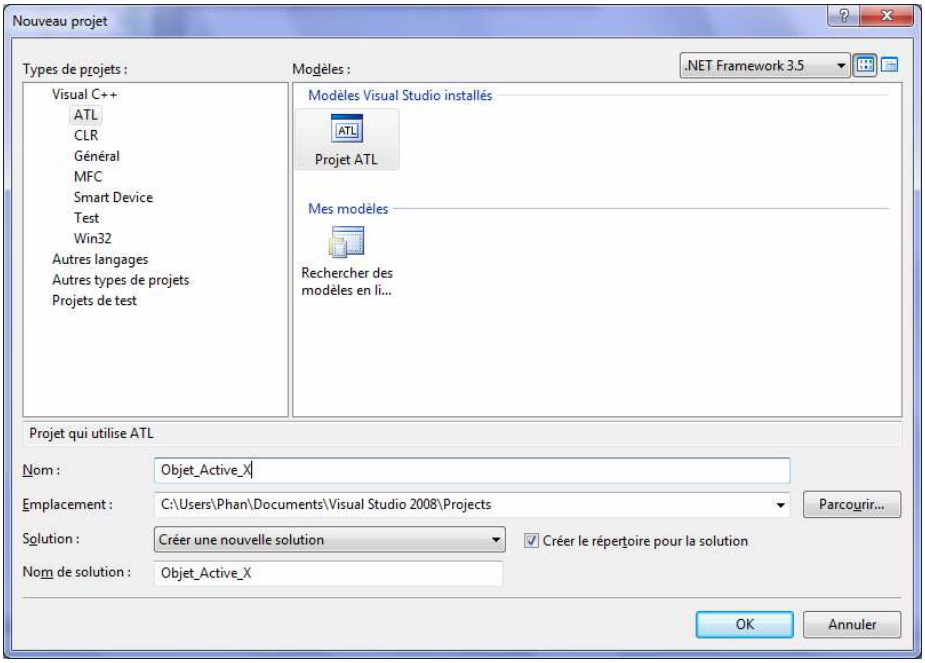
Aller ensuite récupérer le .dll dans le répertoire du projet pour le mettre dans le répertoire système de Windows (Windows/System32/).

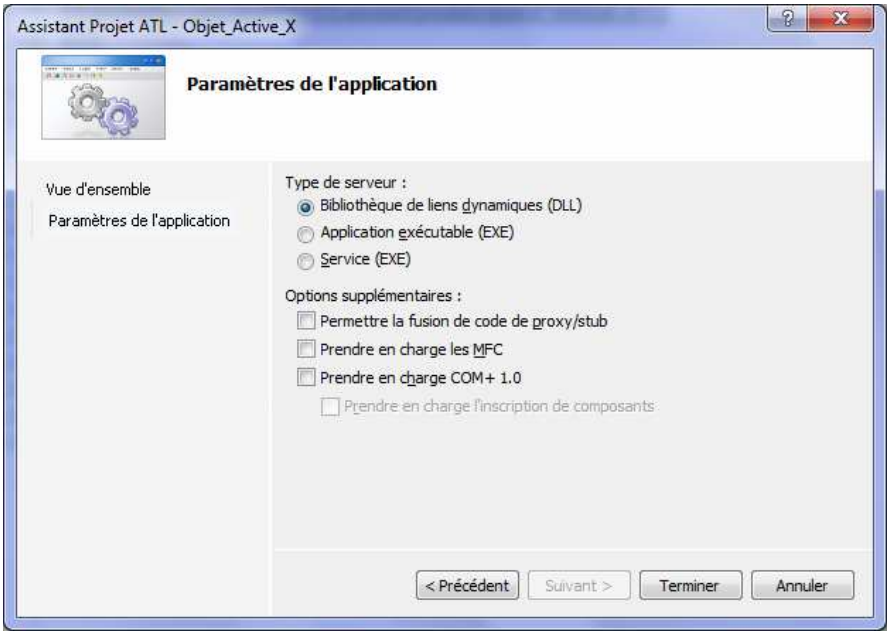


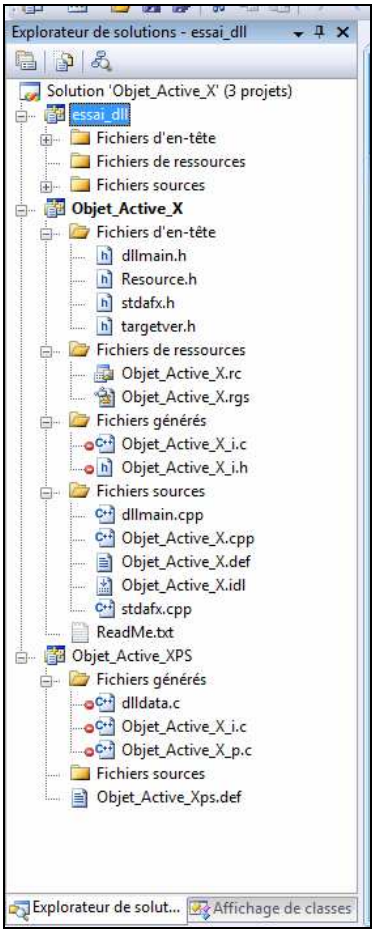
CREATION D'UN OBJET ACTIVEX

Cette partir du tutoriel est inspiré du tutoriel de création d'un objet ActiveX de Raksmey Phan :
http://www.isima.fr/~lacomme/doc/Tuto_VB_ActiveX.pdf

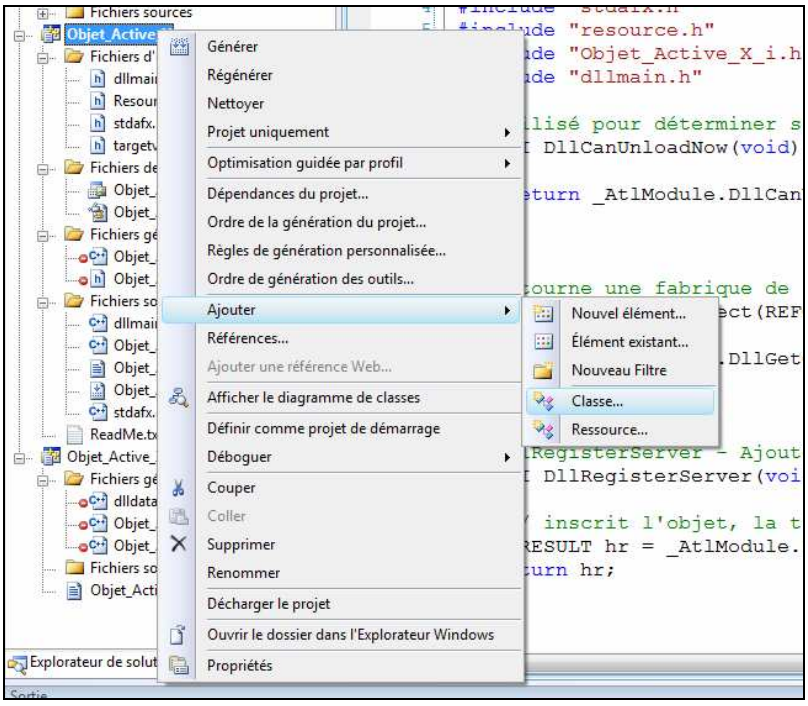
Créer un projet de type « Projet ATL » en Visual C++ ;
Choisir comme Objet_Active_X

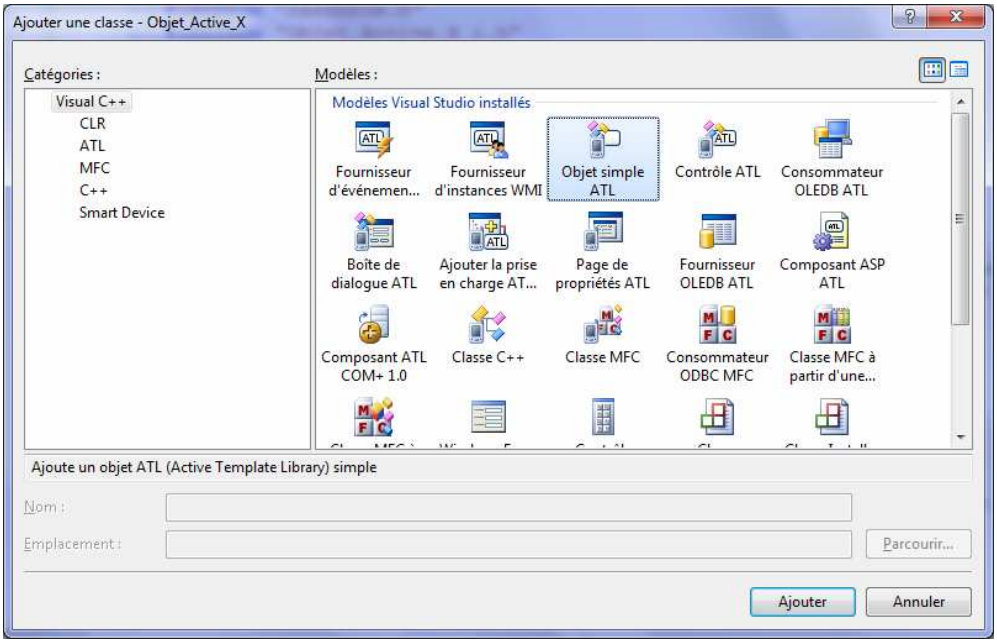




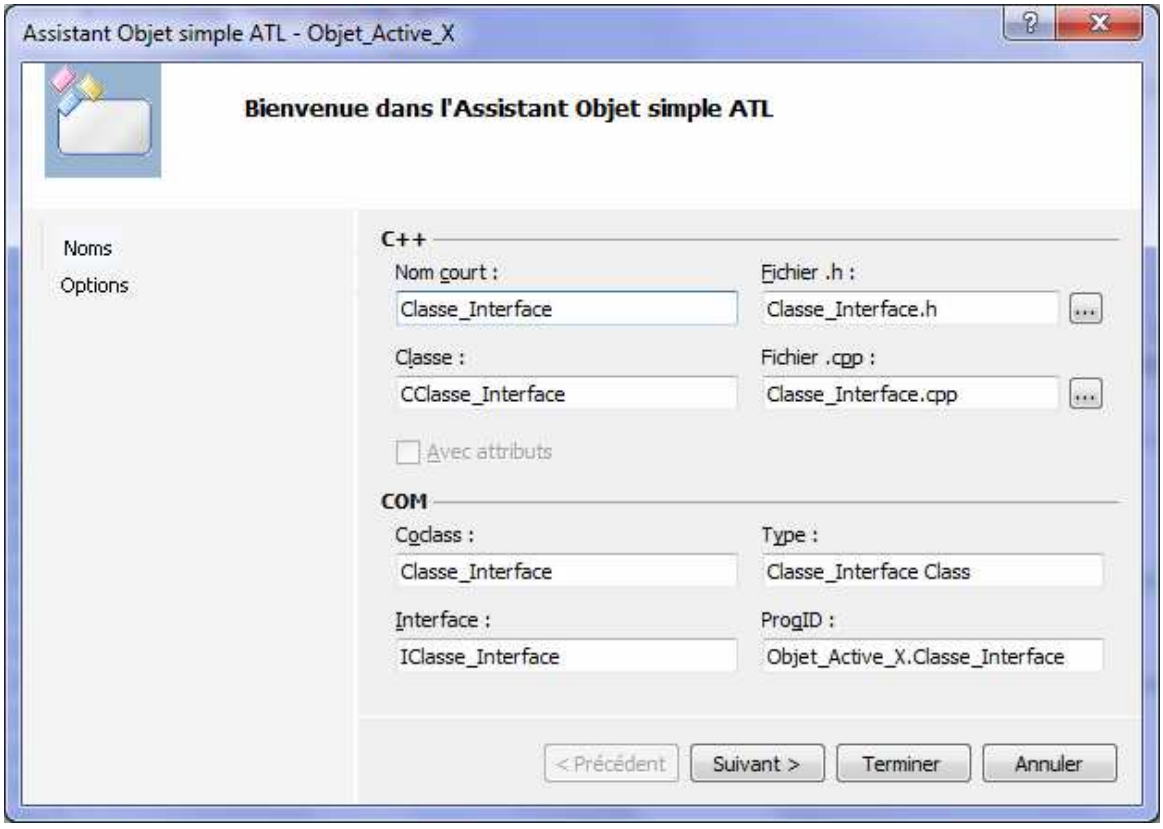


Ajouter une classe à votre ActiveX.



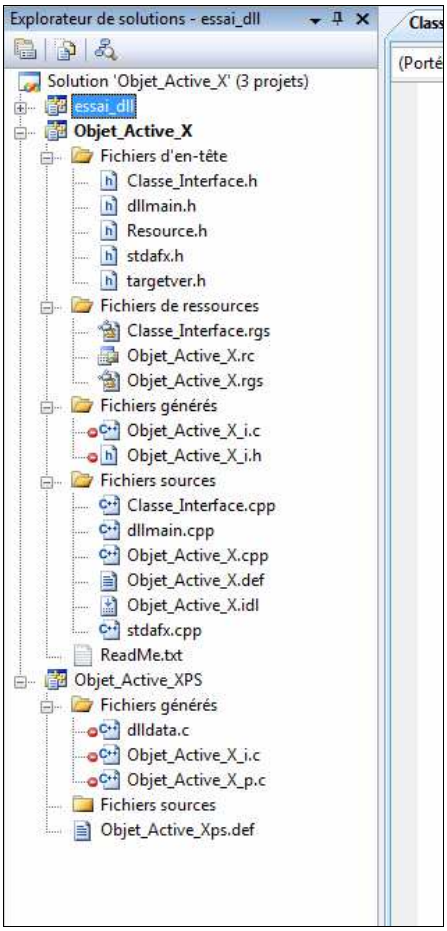


Choisir « Classe_Interface » comme nom pour votre nouvelle classe.

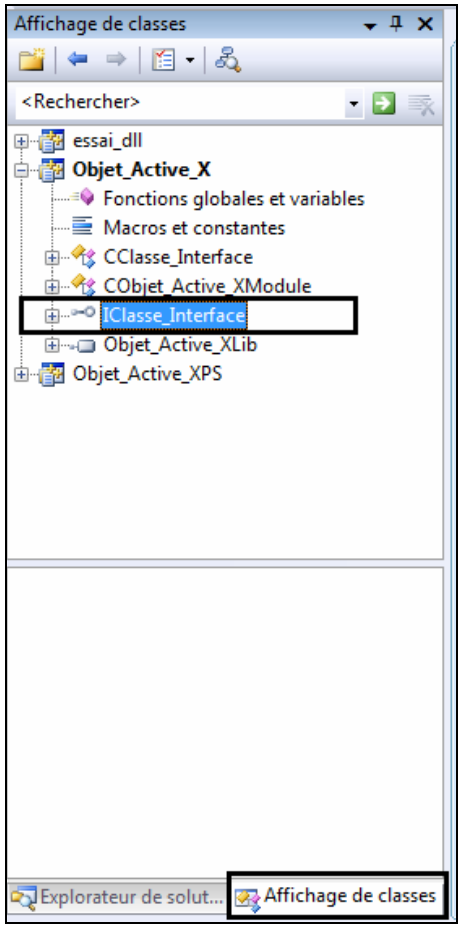


Validez ensuite les autres choix par défaut.

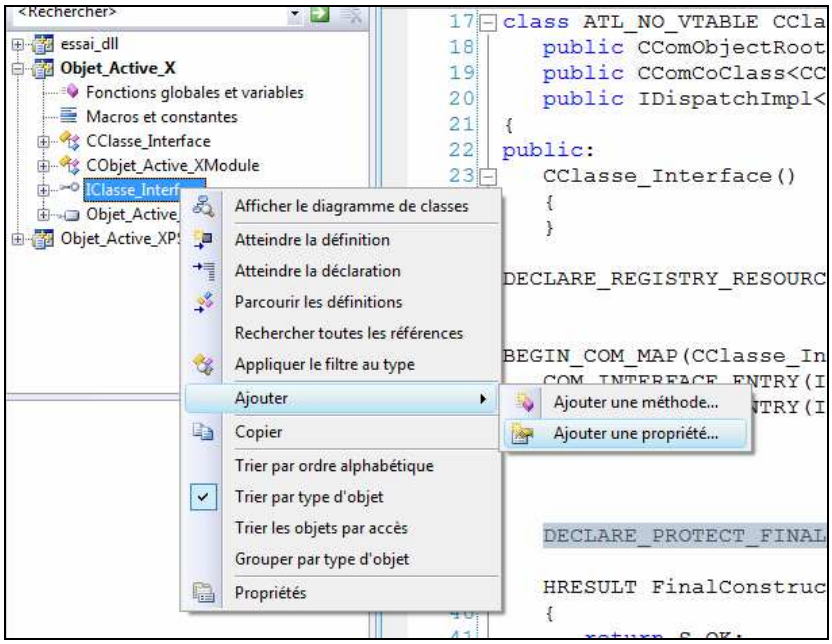
On obtient le projet suivant :



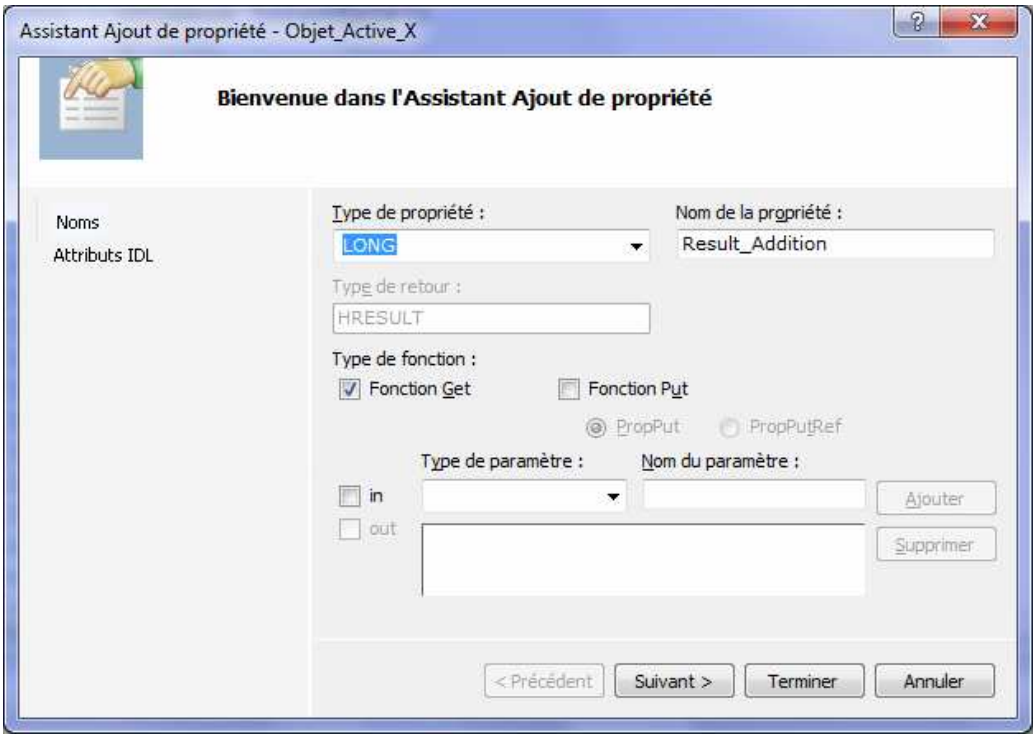
On passe dans l'affichage des classes comme suit :

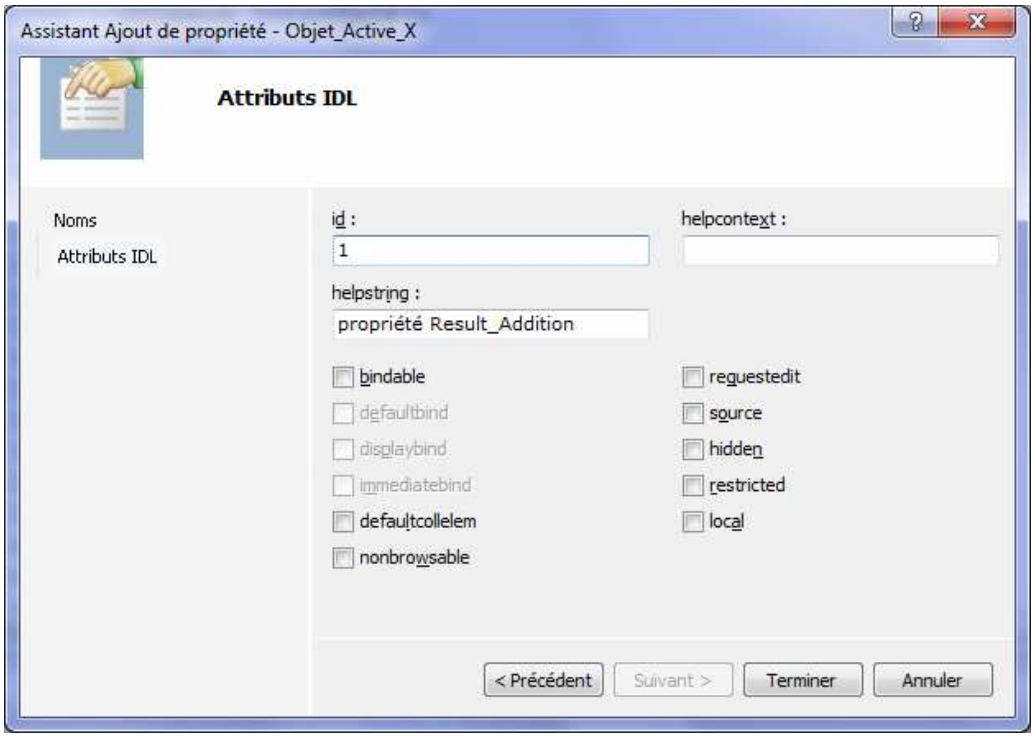


Faire un clic droit sur **IClasse_Interface** pour ajouter une propriété :

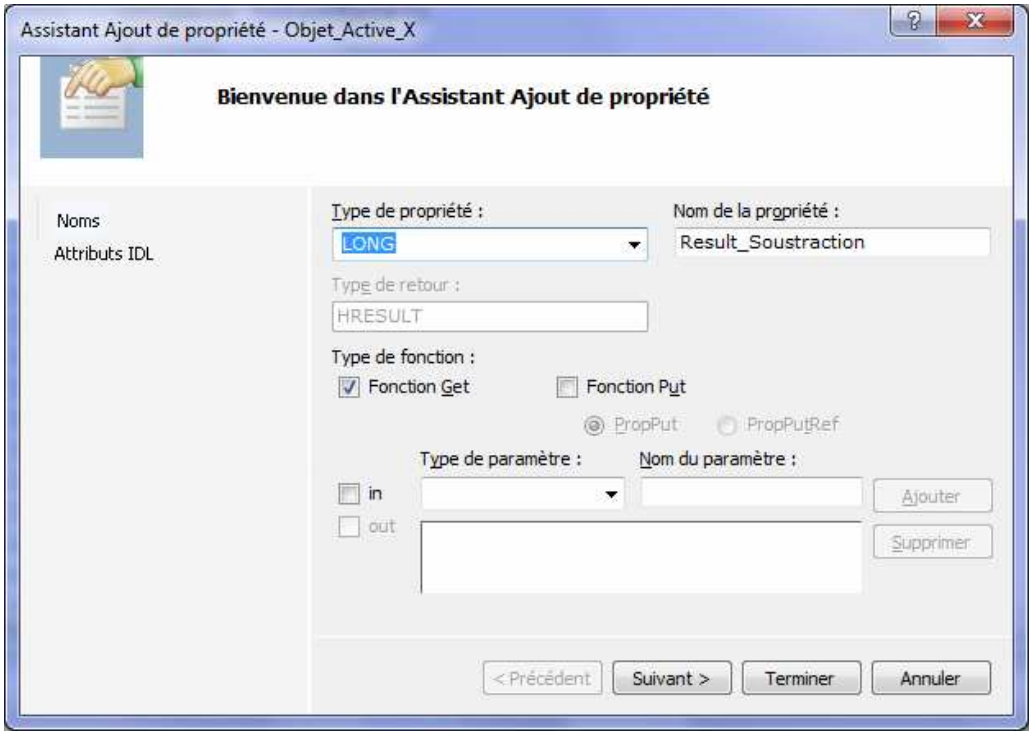


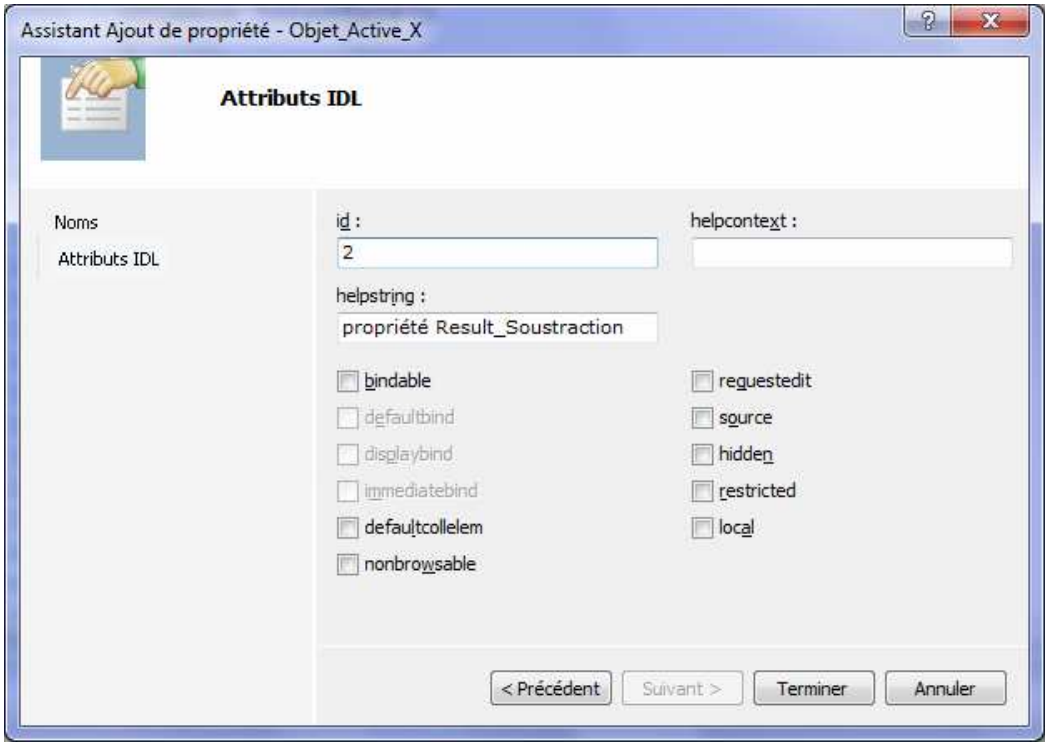
Nous allons déclarer une propriété « Result_Addition » ayant uniquement une méthode get.



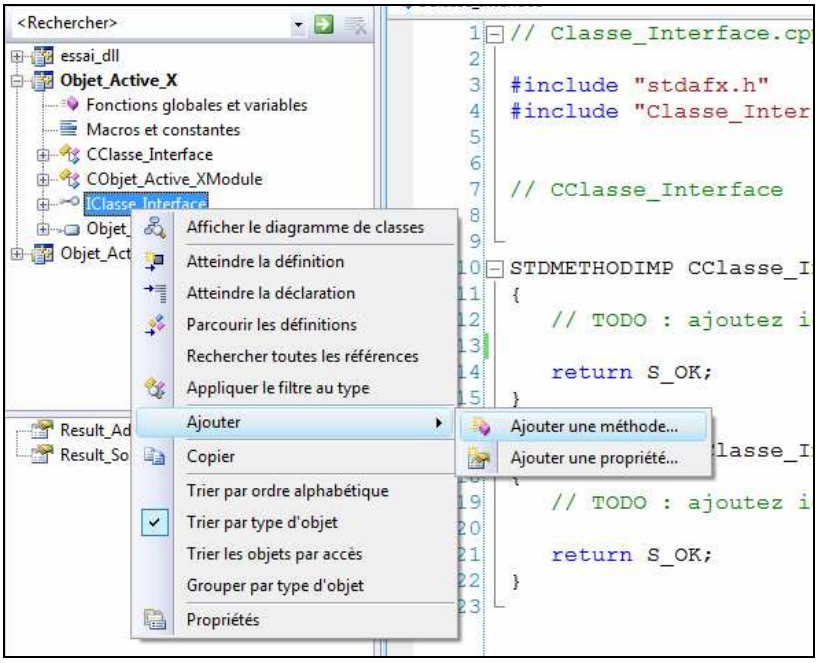


Et on fait de même pour la soustraction.

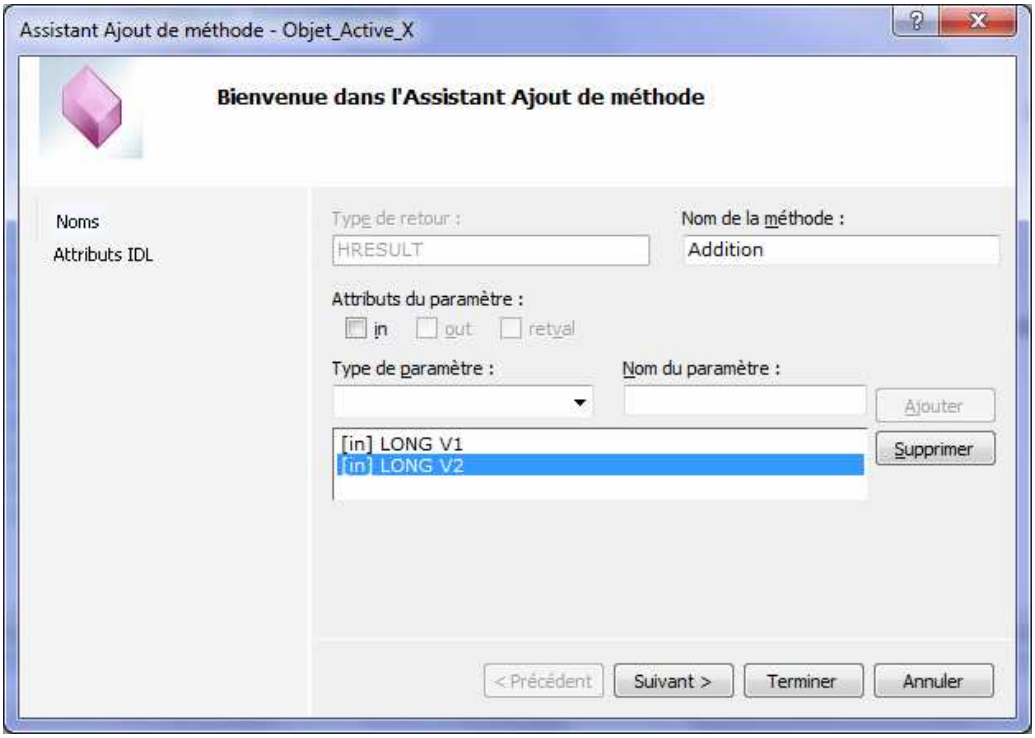


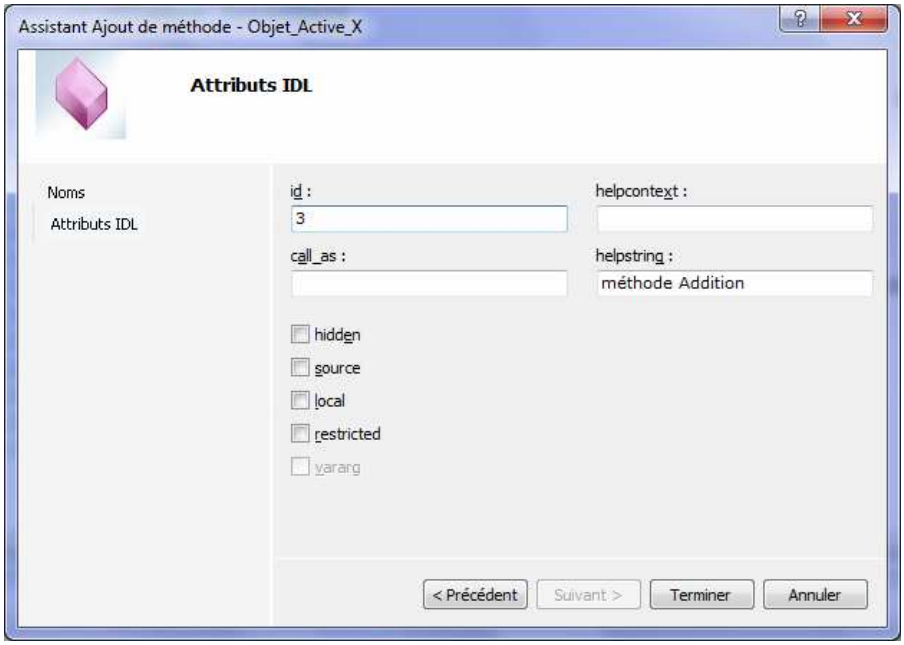


On crée deux nouvelles méthodes pour appeler les fonctions de la dll.

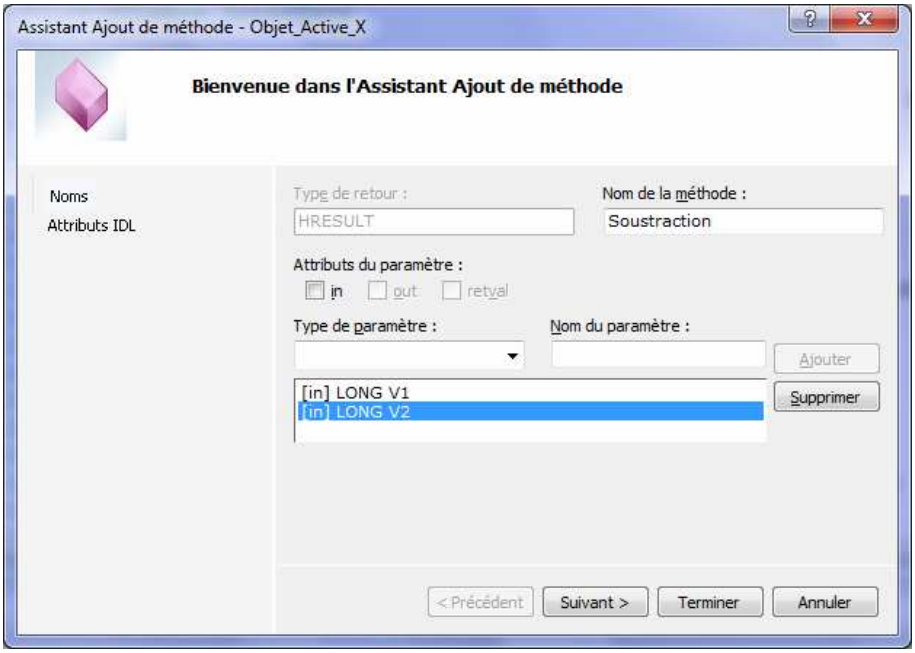


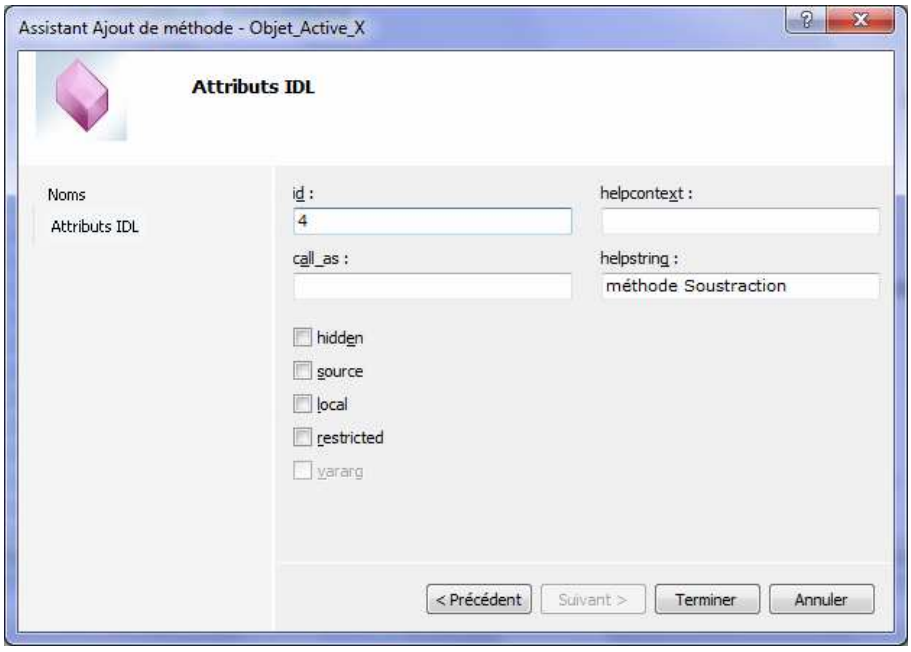
La première s'appelle Addition et comportera deux paramètres V1 et V2 de type LONG.





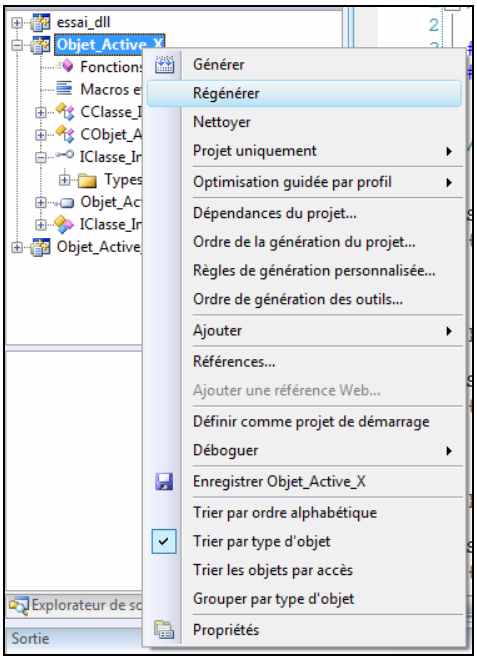
La deuxième s'appelle Soustraction et comportera deux paramètres V1 et V2 de type LONG.



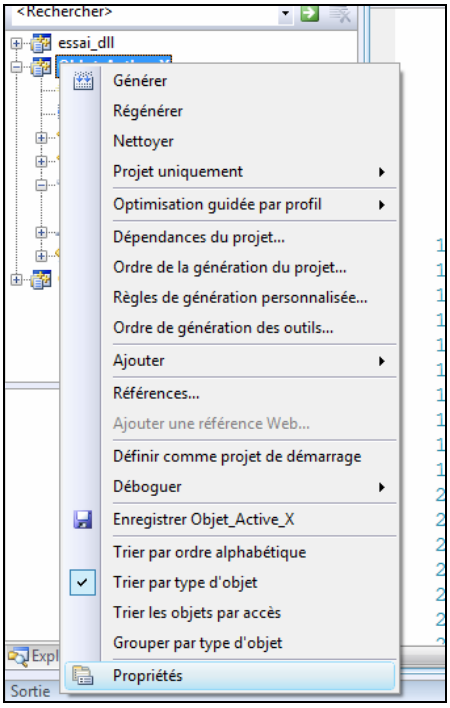


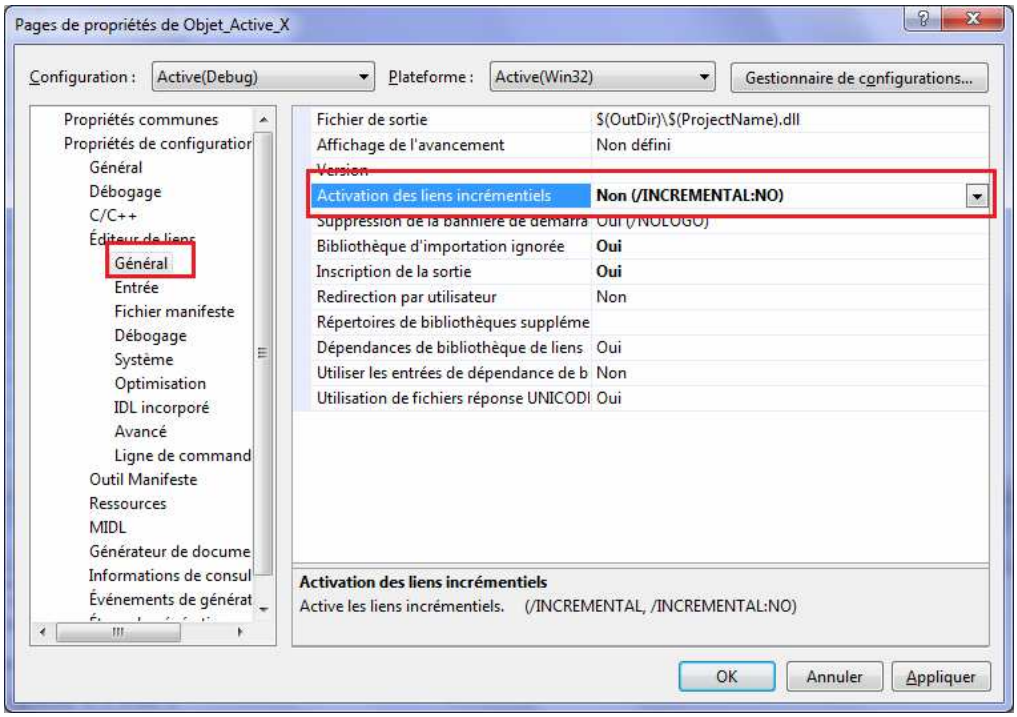
Compiler le projet pour vérifier que tout est en place.

Faire un clic droit sur le projet et choisir Regérer.



En cas d'erreur de « link » il faut modifier les options de compilation.

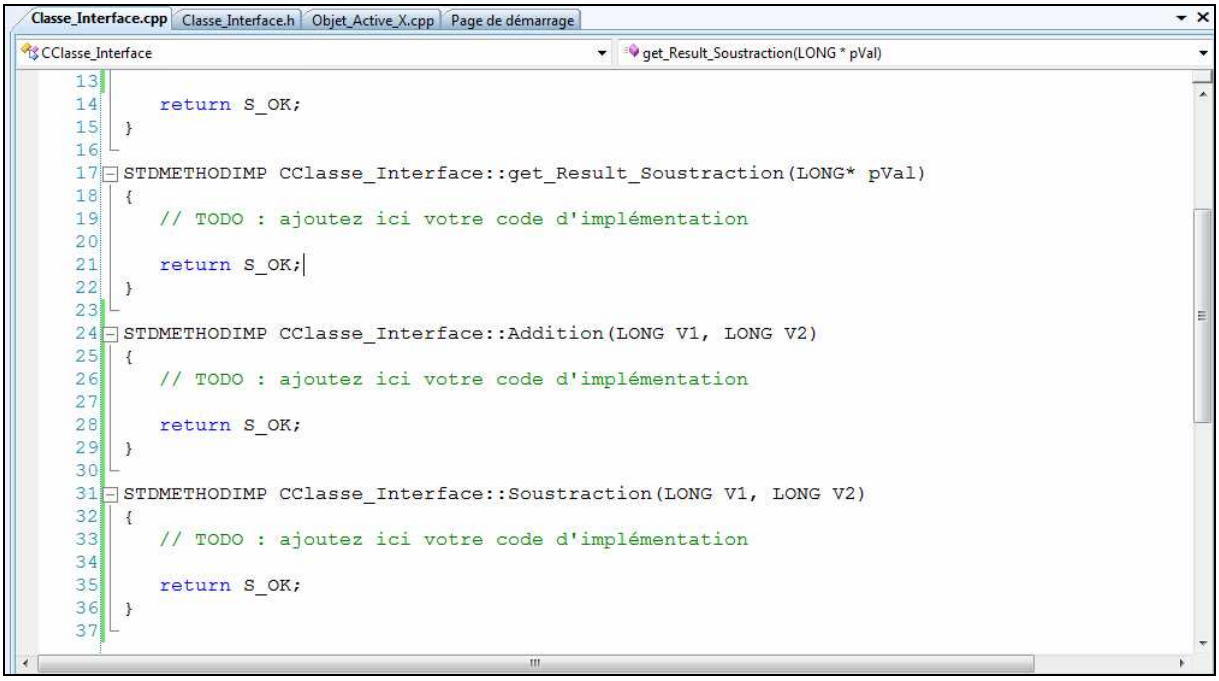

<Rechercher>
essai_dll
Générer
Régénérer
Nettoyer
Projet uniquement
Optimisation guidée par profil
Dépendances du projet...
Ordre de la génération du projet...
Règles de génération personnalisée...
Ordre de génération des outils...
Ajouter
Références...
Ajouter une référence Web...
Définir comme projet de démarrage
Déboguer
Enregistrer Objet_Active_X
Trier par ordre alphabétique
Trier par type d'objet
Trier les objets par accès
Grouper par type d'objet
Propriétés
Sortie


Pages de propriétés de Objet_Active_X
Configuration : Active(Debug)
Plateforme : Active(Win32)
Gestionnaire de configurations...
Propriétés communes
Propriétés de configuration
Général
Débogage
C/C++
Éditeur de liens
Général
Entrée
Fichier manifeste
Débogage
Système
Optimisation
IDL incorporé
Avancé
Ligne de command
Outil Manifeste
Ressources
MIDL
Générateur de docume
Informations de consul
Événements de génerat
Fichier de sortie
$(OutDir)\$(ProjectName).dll
Affichage de l'avancement
Non défini
Version
Activation des liens incrémentiels
Non (/INCREMENTAL:NO)
Bibliothèque d'importation ignorée
Oui
Inscription de la sortie
Oui
Redirection par utilisateur
Non
Répertoires de bibliothèques suppléme
Dépendances de bibliothèque de liens
Oui
Utiliser les entrées de dépendance de b
Non
Utilisation de fichiers réponse UNICODI
Oui
Activation des liens incrémentiels
Active les liens incrémentiels. (/INCREMENTAL, /INCREMENTAL:NO)
OK
Annuler
Appliquer

Appeler les fonctions de la dll.

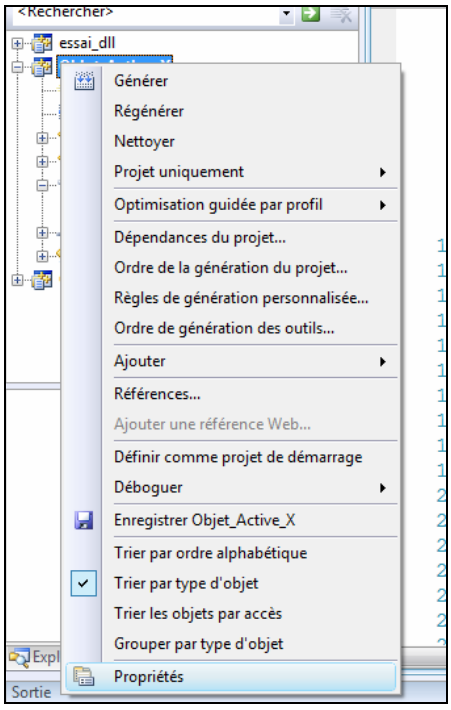
Ouvrir le fichier Classe_Interface.cpp.

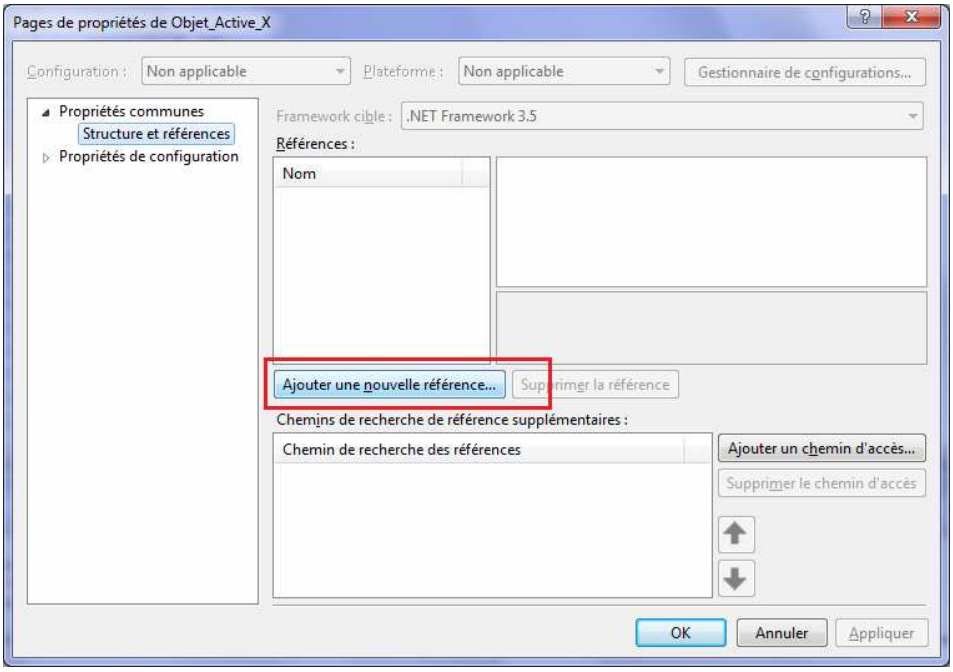
Les deux fonctions crées précédemment sont dans ce fichier.

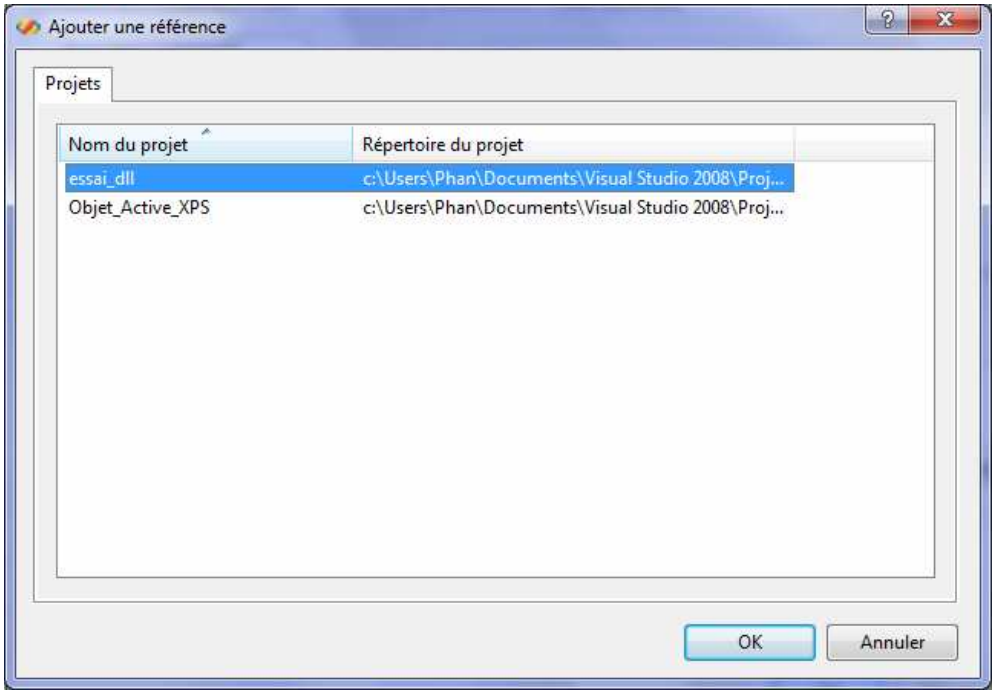


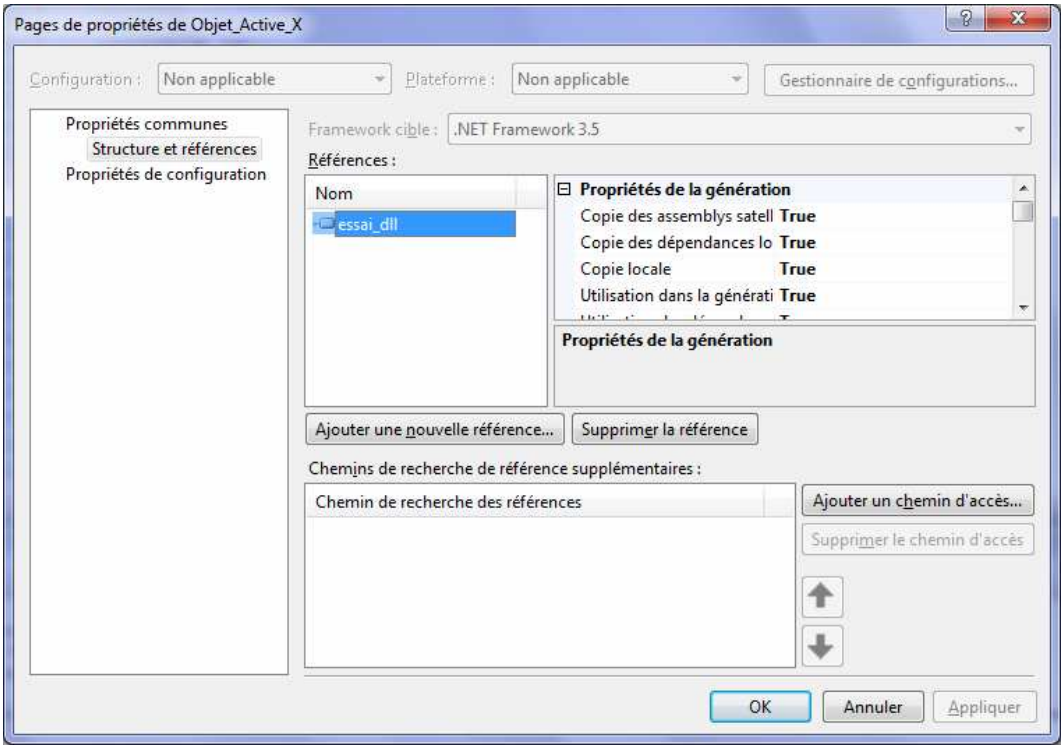
Il faut maintenant inclure un lien vers la dll.

Pour cela, faire un clic droti sur Object_Active_X

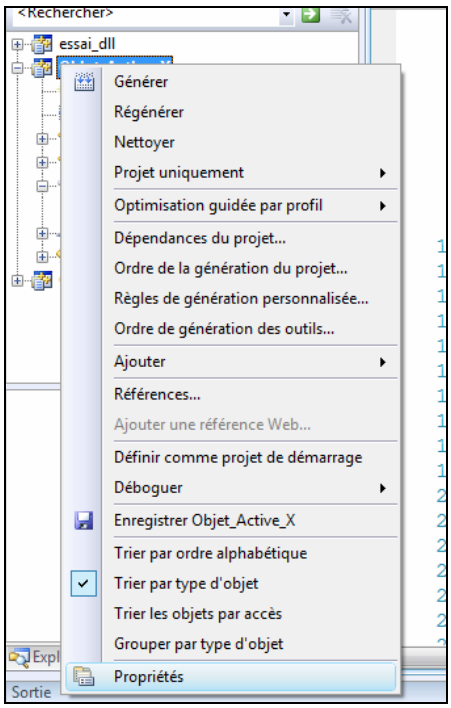


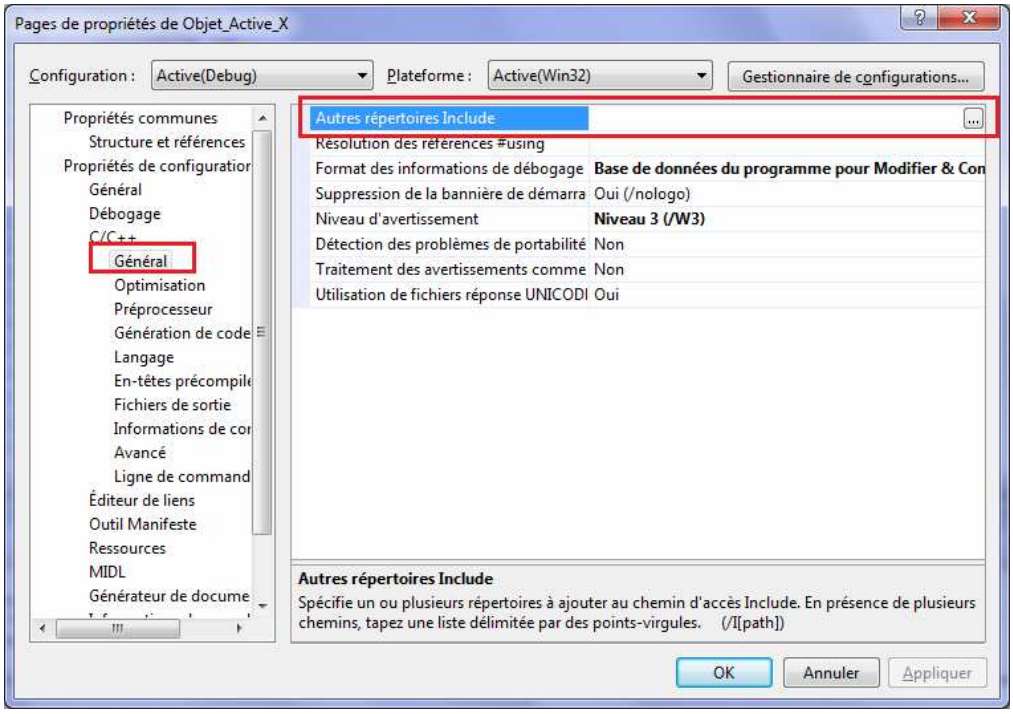

Pages de propriétés de Objet_Active_X
Configuration : Non applicable
Plateforme : Non applicable
Gestionnaire de configurations...
Propriétés communes
Structure et références
Propriétés de configuration
Framework cible : .NET Framework 3.5
Références :
Nom
Ajouter une nouvelle référence...
Supprimer la référence
Chemins de recherche de référence supplémentaires :
Chemin de recherche des références
Ajouter un chemin d'accès...
Supprimer le chemin d'accès
OK
Annuler
Appliquer

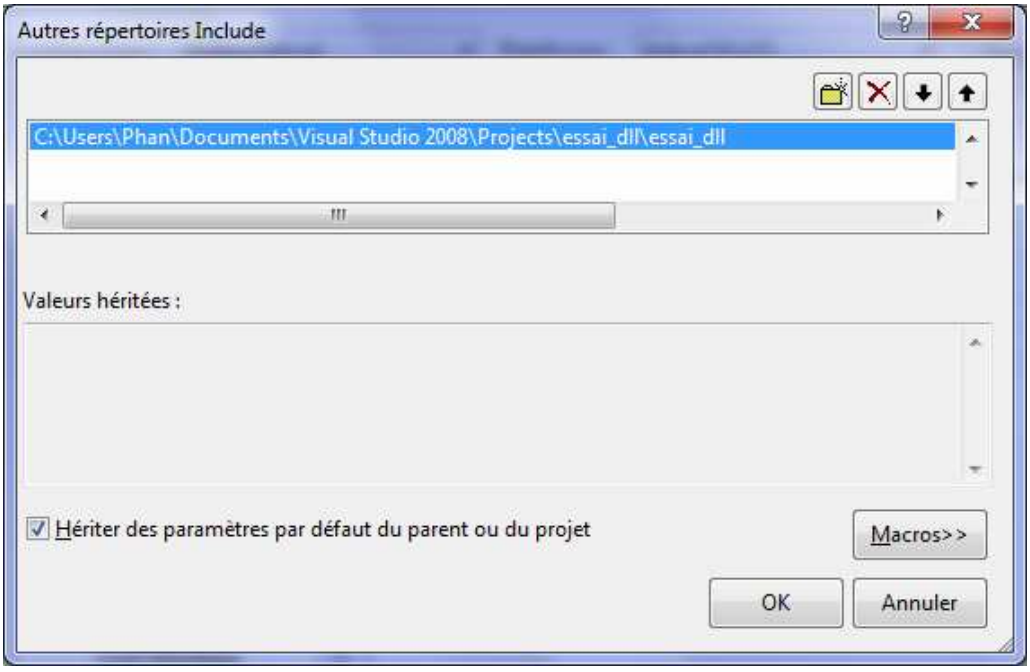

Ajouter une référence
Projets
Nom du projet
Répertoire du projet
essai_dll
c:\Users\Phan\Documents\Visual Studio 2008\Proj...
Objet_Active_XPS
c:\Users\Phan\Documents\Visual Studio 2008\Proj...
OK
Annuler



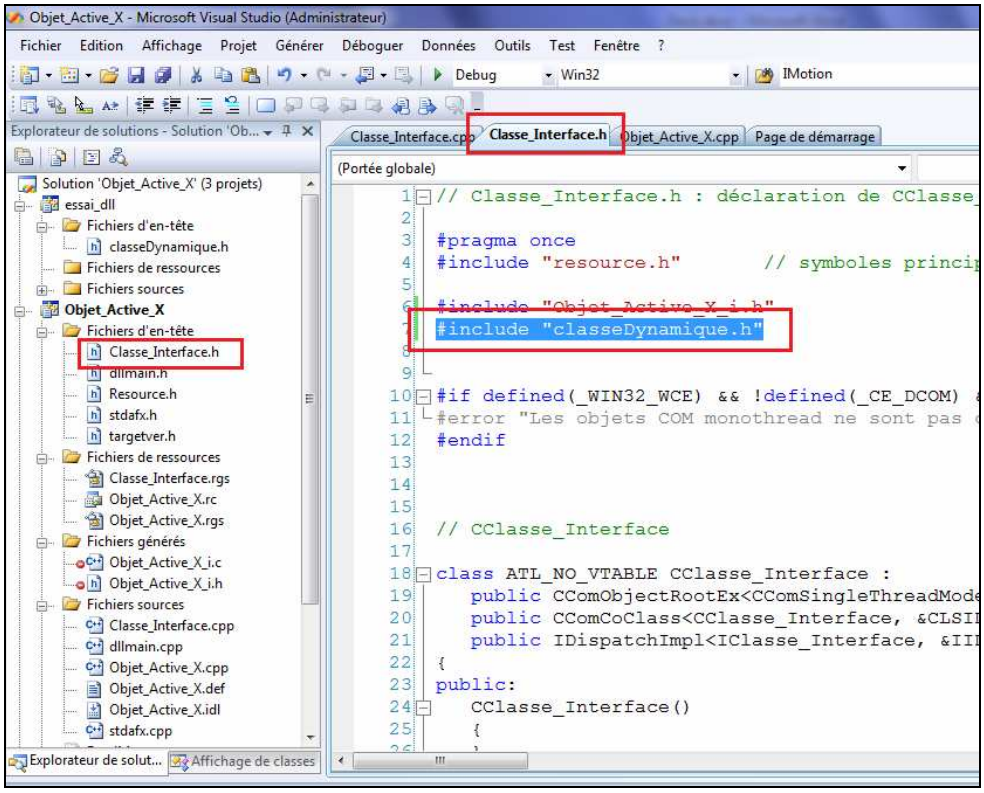
Il faut inclure un lien vers le répertoire de la dll.





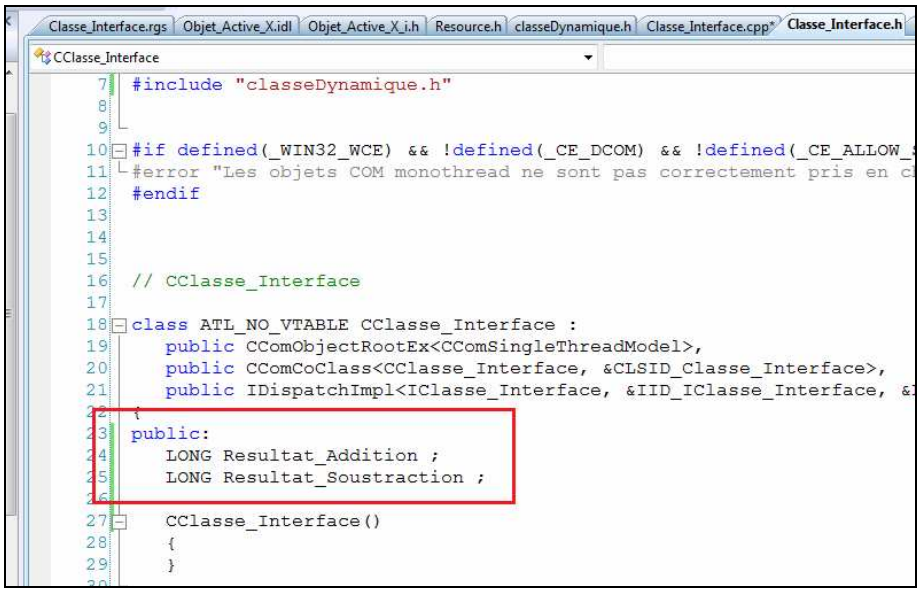


Il faut mettre à jour le fichier « Classe_Interface.h »



Mettre à jour Classe_Interface.h.

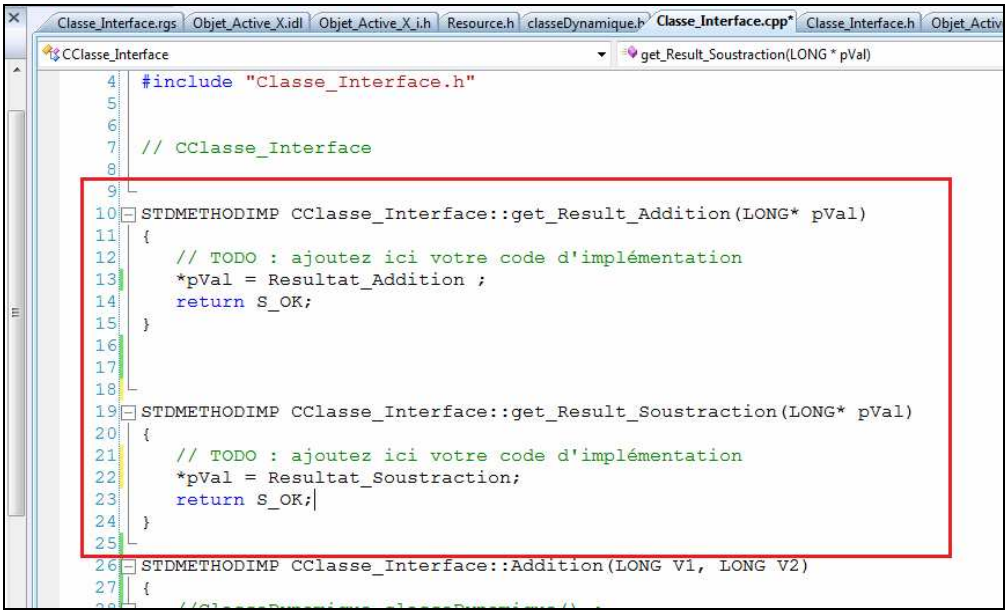
Ajouter la ligne suivante comme variable de l'objet CClasse_Calcul : cette variable nous permet de stocker les valeurs dans l'objet.



Mettre à jour le fichier Classe_Interface.cpp.

Modifier les méthodes comme suit :

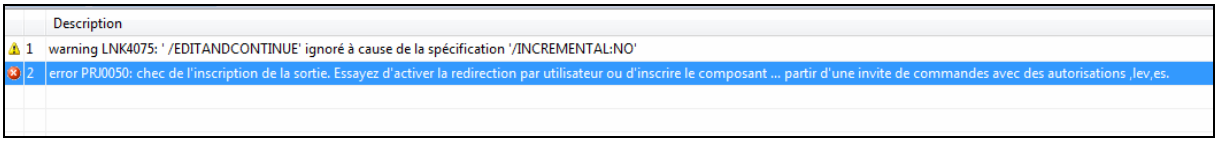
```
STDMETHODIMP CClasse_Interface::get_Result_Addition(LONG* pVal)
{
   // TODO : ajoutez ici votre code d'implémentation
   *pVal = Resultat_Addition ;
   return S_OK;
}

STDMETHODIMP CClasse_Interface::get_Result_Soustraction(LONG* pVal)
{
   // TODO : ajoutez ici votre code d'implémentation
   *pVal = Resultat_Soustraction;
   return S_OK;
}
```

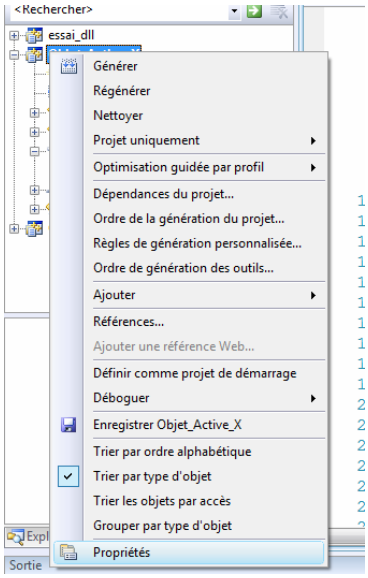


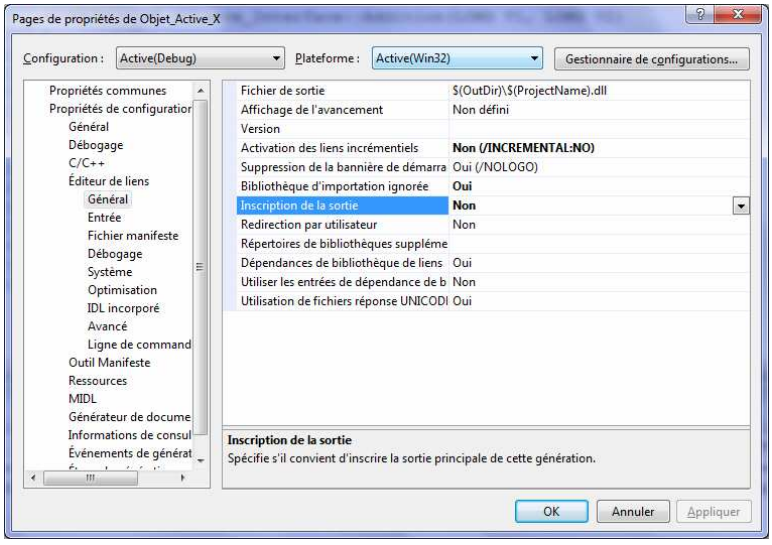
Soustraction : appel de la .dll.

```
STDMETHODIMP CClasse_Interface::Soustraction(LONG V1, LONG V2)
{
   Resultat_Soustraction = Soustraction_Externe(V1, V2);

   return S_OK;
}
```

A la compilation, si vous rencontrez des problèmes de type « Echec inscription de sortie » modifier les propriétés comme suite.







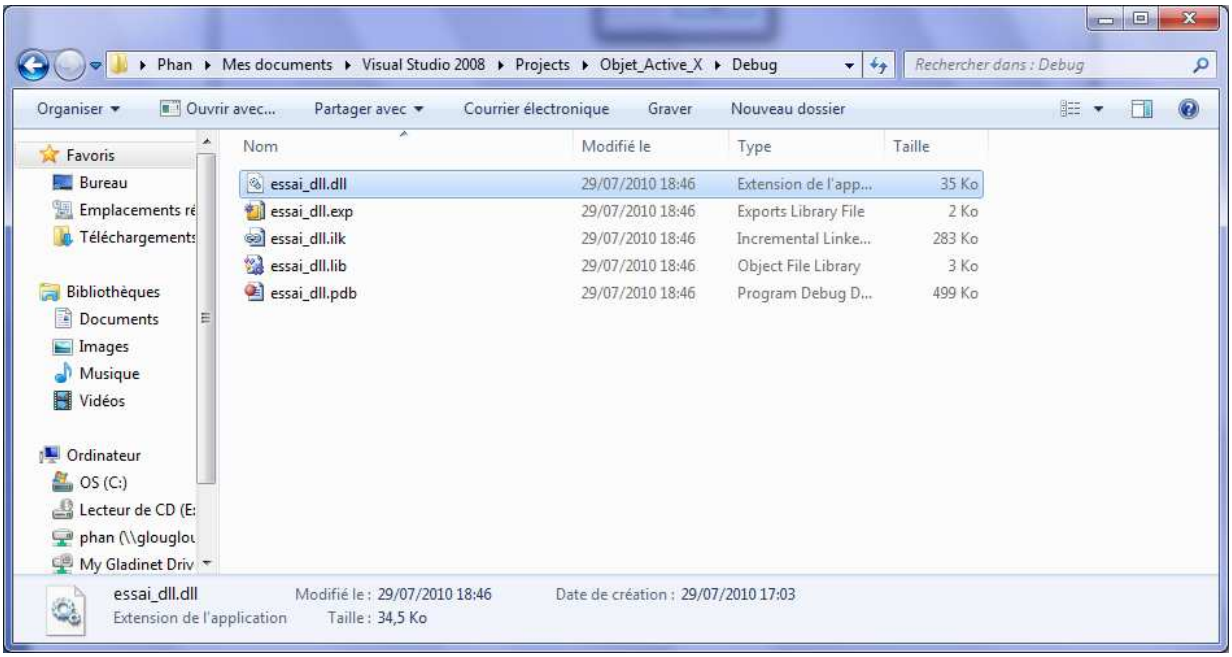
Addition test de la dll.

```
STDMETHODIMP CClasse_Interface::Addition(LONG V1, LONG V2)
{
   ClasseDynamique classeDynamique() ;
   Resultat_Addition = classeDynamique.Addition_Interne(V1, V2) ;
   return S_OK;
}
```

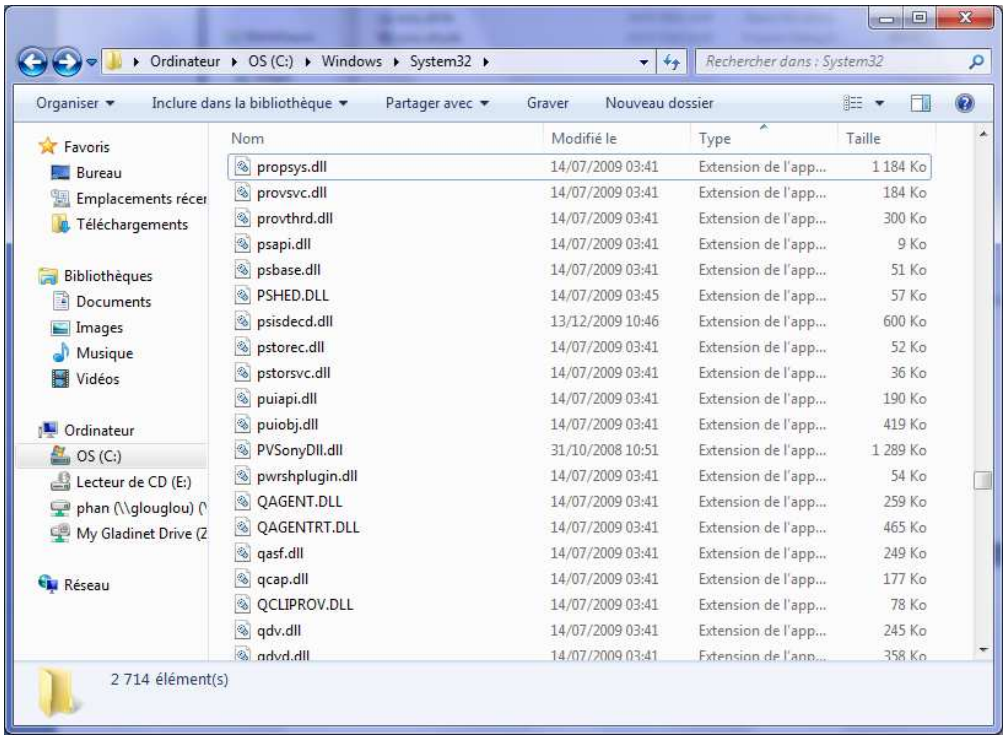
TEST D'UN ACTIVEX AVEC JAVASCRIPT

Étape 1 :

Copier la dll essai_dll.dll se trouvant dans le répertoire du projet Objet_Active_X : …\Objet_Active_X\Debug. Attention à ne pas confondre avec le même essai_dll.dll qui se trouve dans le répertoire du projet Essai_dll.



Coller ce essai_dll.dll dans …/Windows/System32/.



Étape 2 :

Ouvrir un editeur de texte pour créer une page html et mettre l'exemple suivant.

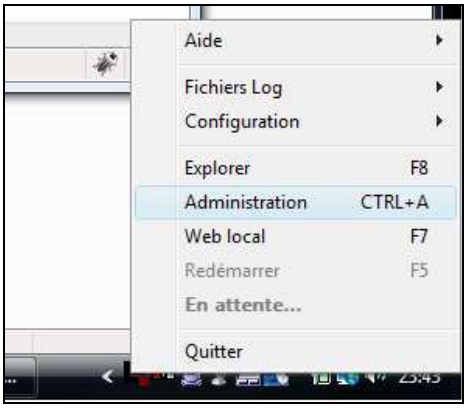
```
<html>

   <!-- En-tête de la page HTML -->
   <head>

      <!-- Déclaration de code JavaScript -->
      <script type="Text/JavaScript">
         function start()
         {
            // Création de l'objet ActiveX
            var obj = new ActiveXObject("Objet_Active_X.Classe_Interface");

            // Appel à la méthode qui fait le calcul
            obj.Addition(8,15) ;
            // Récupération du résultat
            alert(obj.Resultat_Addition);

            // Appel à la méthode qui fait le calcul
            obj.Soustraction(15, 8) ;
            // Récupération du résultat
            alert(obj.Resultat_Soustraction);
         }
      </script>

   </head>


   <!—Le corps de la page HTML -->
   <body onload="start()">
   </body>

</html>
```

TEST D'UN ACTIVEX A PARTIR DE PHP

Étape 1 :

Installer un serveur php. Par exemple : http://www.easyphp.org/

Vérifier que easyphp fonctionne en vous connectant sur la console d'administration.



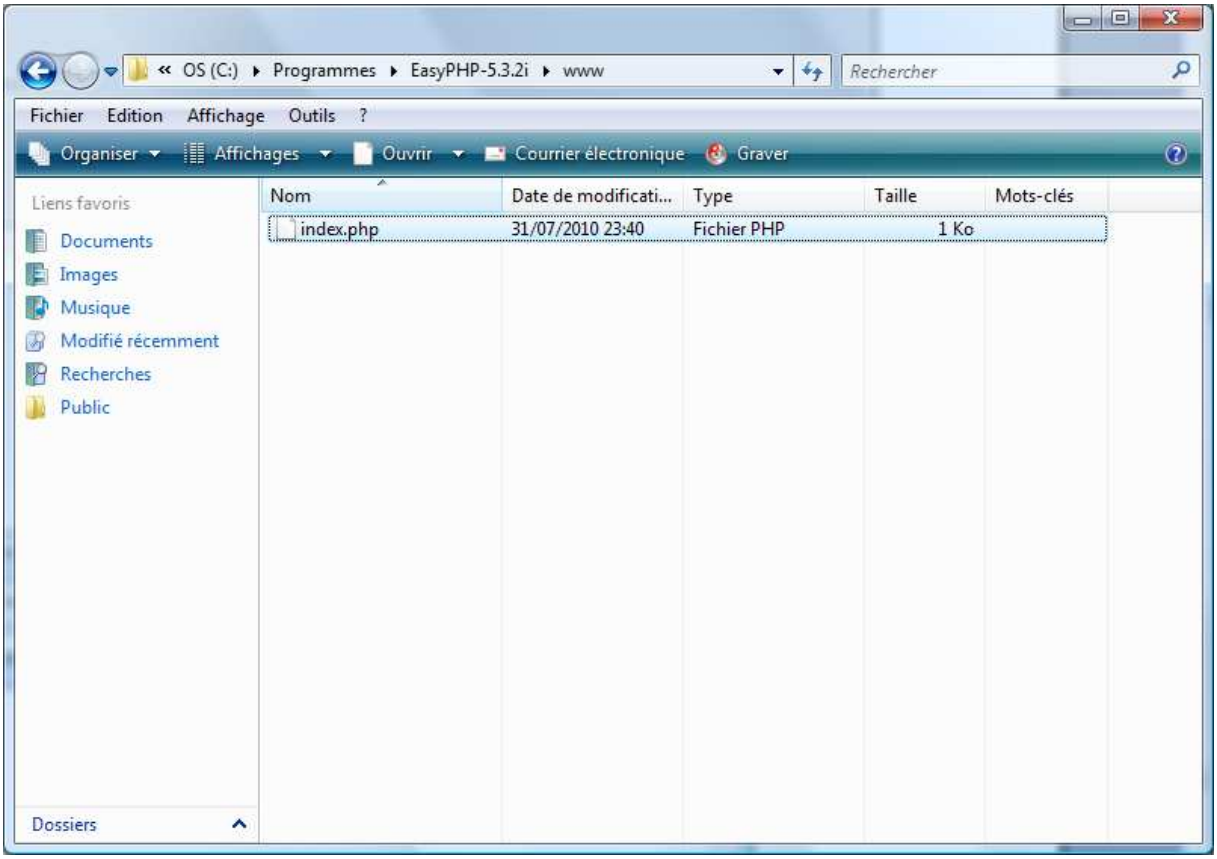
Vous devez obtenir une page comme celle-ci :

Étape 2 :

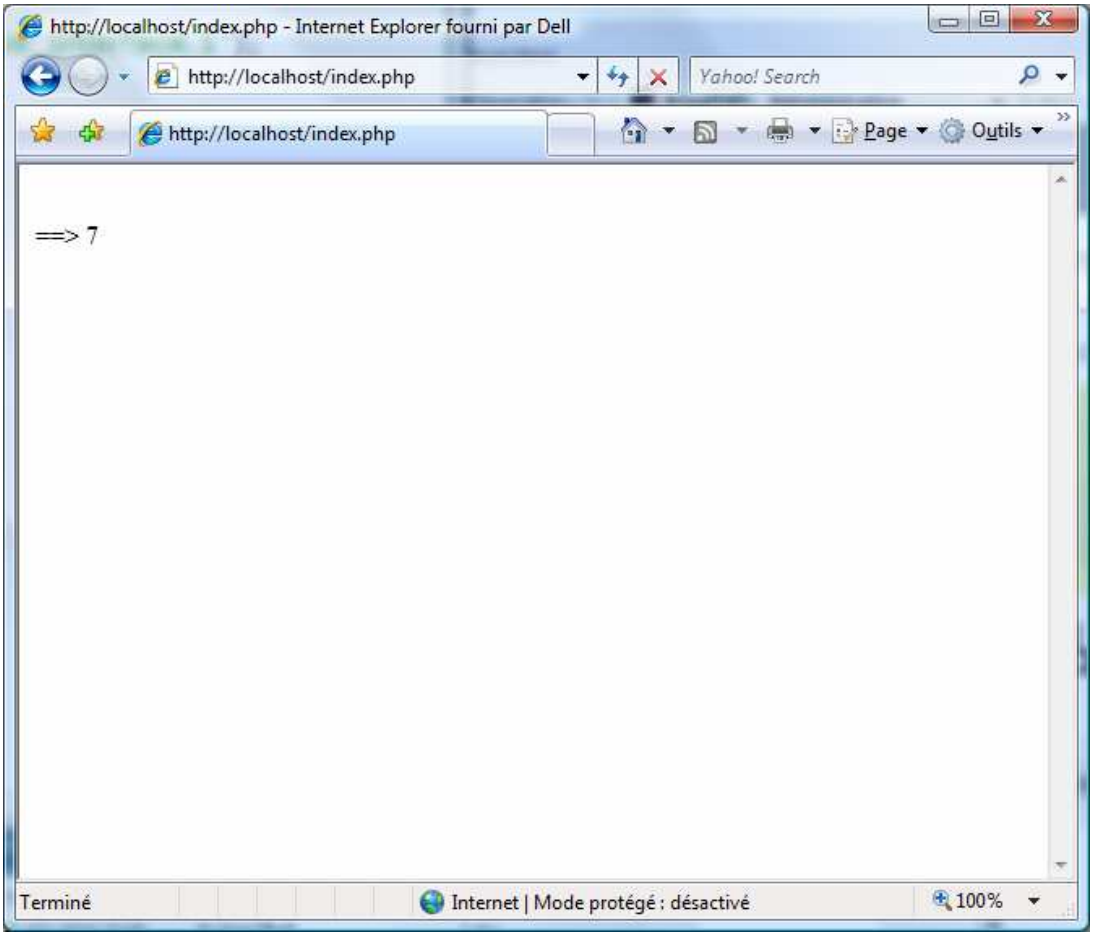
Avec un éditeur de texte, taper le texte suivant dans un fichier nommé index.php.

```
<?php

   $obj2 = new COM("Objet_Active_X.Classe_Interface.1");
   $b=new VARIANT(1);
   $c=new VARIANT(3);
   $obj2->Addition(3,4);

   echo $b."<br>----------";


   $e= $obj2-> Resultat_Addition;

  echo $e;


?>
```

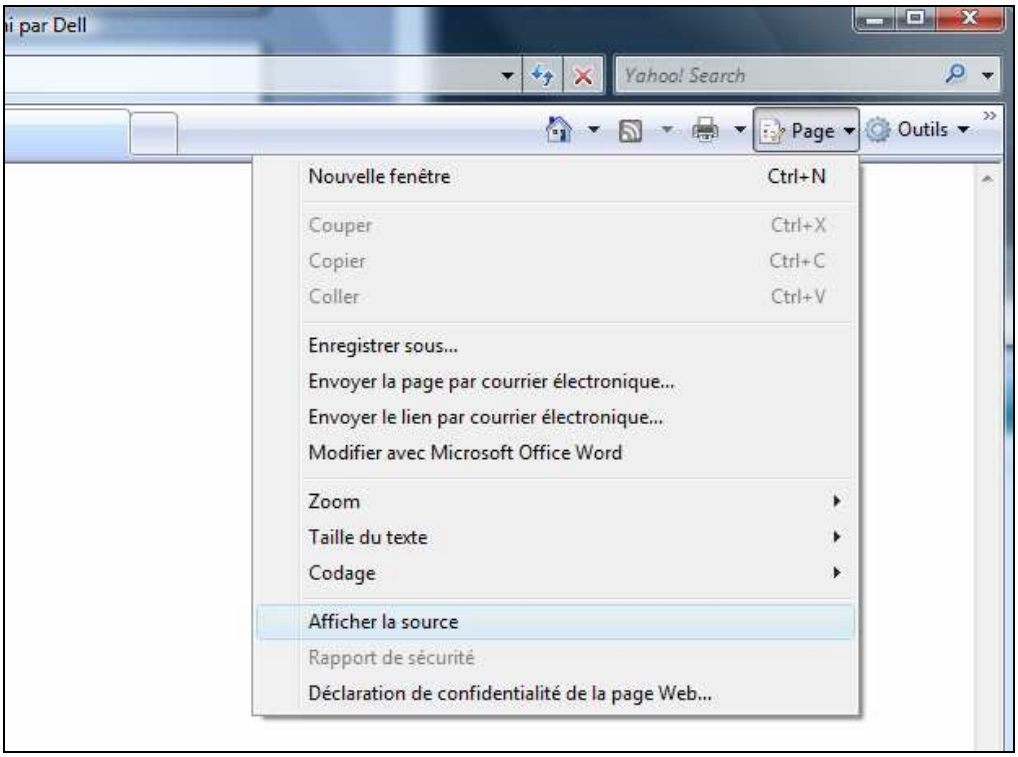
Déposer ce fichier dans le sous répertoire **www** de **easyphp**.



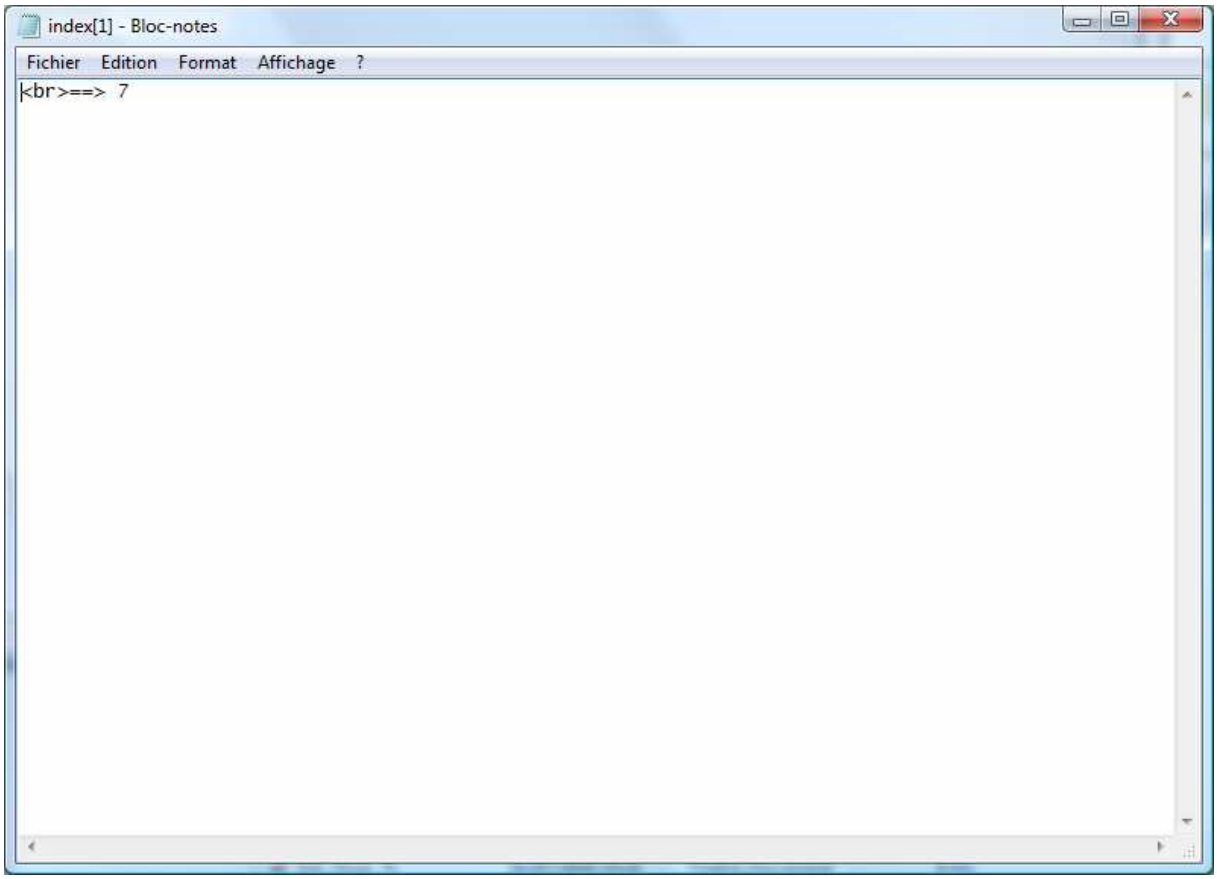
A l'aide d'Internet Explorer connectez vous sur http://localhost/index.php



Dans l'onglet Page, vous pouvez accéder au code source de la page.



On constate alors que le code source de la page contient uniquement
==> 7, ce qui montre que le code php est interprété par le serveur.



---------- FIN ----------