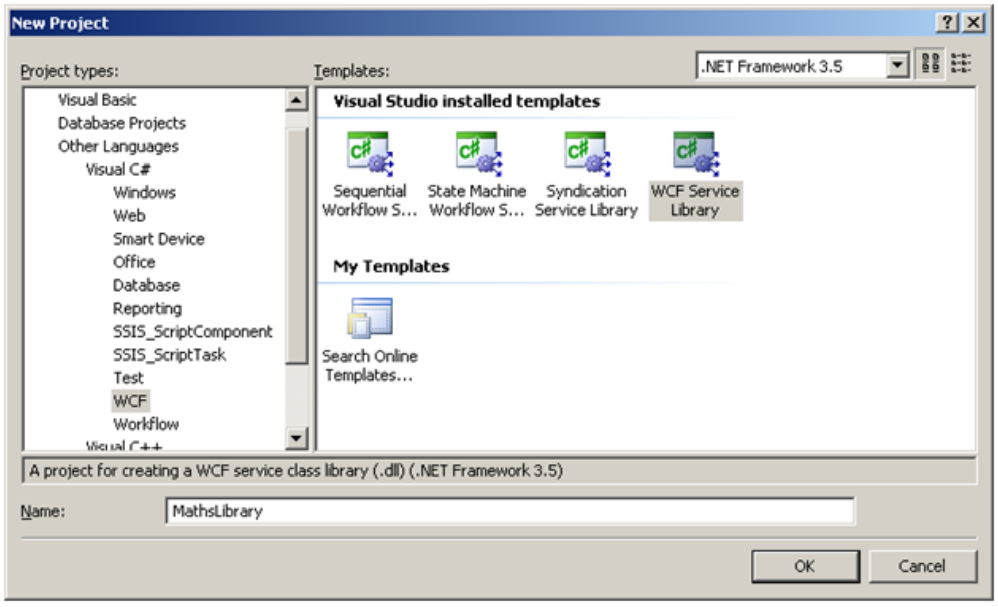
Applications Réparties
Services WCF (Windows Communication Foundation)

1 Création de services WCF avec C

1.1 Création de Service WCF de base en C #. Net

WCF (Windows Communication Foundation) est une partie de. NET. Nous utiliserons Visual Studio 2012 ou une version postérieur pour ce TD.

Le projet à créer est un New Project -> WCF -> WCFClassLibrary projet comme indiqué ci-dessous



Donner le nom **MathsLibrary** à votre projet.

Le projet contient un échantillon des fichiers Service1.cs & IService.cs.

Nous allons les effacer pour ajouter notre propre service.

Faites un clic droit sur MathsLibrary -> Ajouter -> New Item -> Select Class1.cs.

Renommez-le MathsOperations.cs

Cela va créer un fichier de classe simple. Ouvrez le fichier et rendez la classe publique.

De la même façon ajouter une classe IMathsOperations.cs qui sera une interface, qui fournira une liste de toutes les opérations que le service WCF propose.

Ouvrez IMathsOperations.cs et changer pour **public IMathsOperations**

Ajoutez le code ci-dessous.

Applications Réparties
Services WCF (Windows Communication Foundation)

```
namespace MathsLibrary
{
    public interface IMathsOperations
    {
        int Add(int num1, int num2);
        int Multiply(int num1, int num2);
    }
}
```

Pour que IMathsOperations devienne un contrat de service WCF, ajouter un attribut **[ServiceContract]**.

Ainsi toute opération que vous souhaitez rendre visible pour le client doit être décoré avec l'attribut **[OperationContract].**

[ServiceContract] et **[OperationContract]** sont inclus dans l'espace de nom System.ServiceModel

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.ServiceModel;

namespace MathsLibrary
{
    [ServiceContract]
    public interface IMathsOperations
    {
        [OperationContract]
        int Add(int num1, int num2);
        [OperationContract]
        int Multiply(int num1, int num2);
    }
}
```

Maintenant, une fois le contrat fixé, nous pouvons implémenter cette interface dans notre service comme ci-dessous

Applications Réparties
Services WCF (Windows Communication Foundation)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace MathsLibrary
{
    public class MathsOperations : IMathsOperations
    {


        public int Add(int num1, int num2)
        {
            return num1 + num2;
        }

        public int Multiply(int num1, int num2)
        {
            return num1 * num2;
        }


    }
}
```

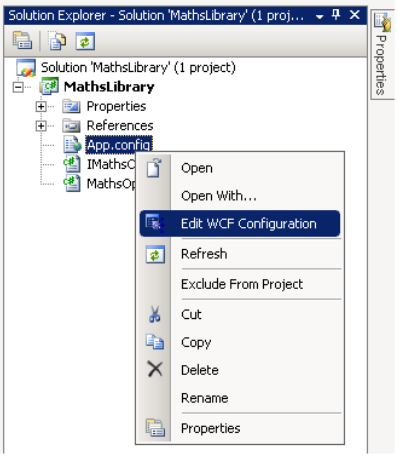
Générez le projet une fois que vous avez terminé à cela.

Ajoutez ou modifier App.config dans la solution. App.config contient des détails sur les endpoints, ce qui comprend les paramètres ABC (address et binding principalement).

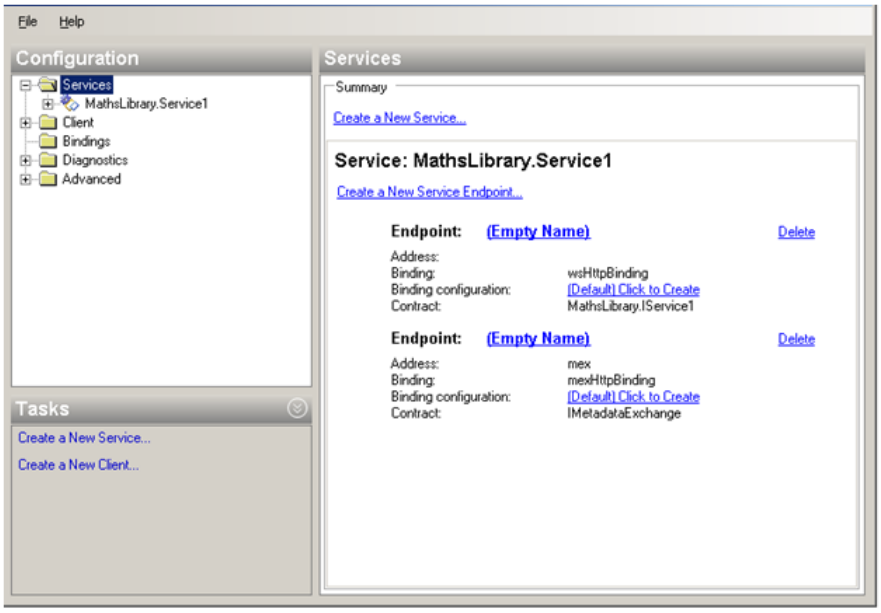
- Address est l'adresse Où le service peut être trouvé.
- Binding est le Comment accéder aux services
- Contrat est ce que (Quoi) contient le service contient.

Maintenant, nous allons modifier App.config. Faites un clic droit sur App.config, cliquez sur Modifier la configuration WCF

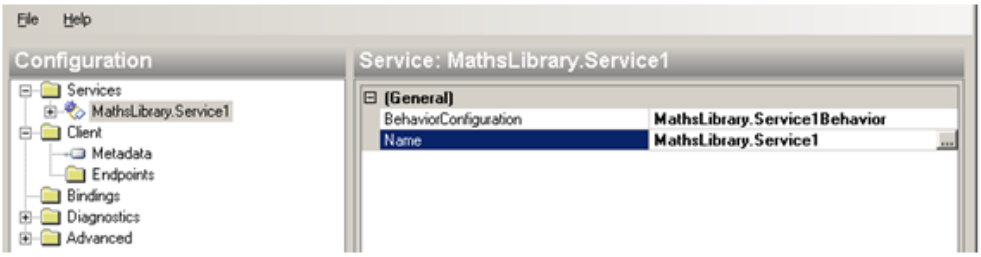
Applications Réparties
Services WCF (Windows Communication Foundation)



Le popup ci-dessous doit apparaître



Sélectionnez le Service1 du panneau de gauche, la fenêtre suivante doit apparaître

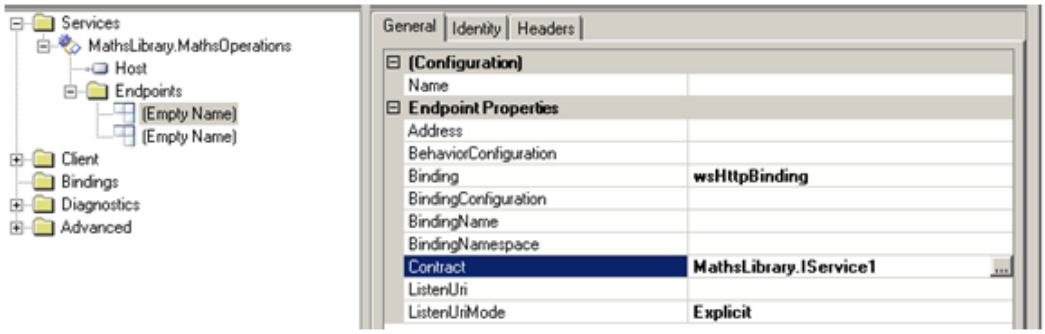


Applications Réparties
Services WCF (Windows Communication Foundation)

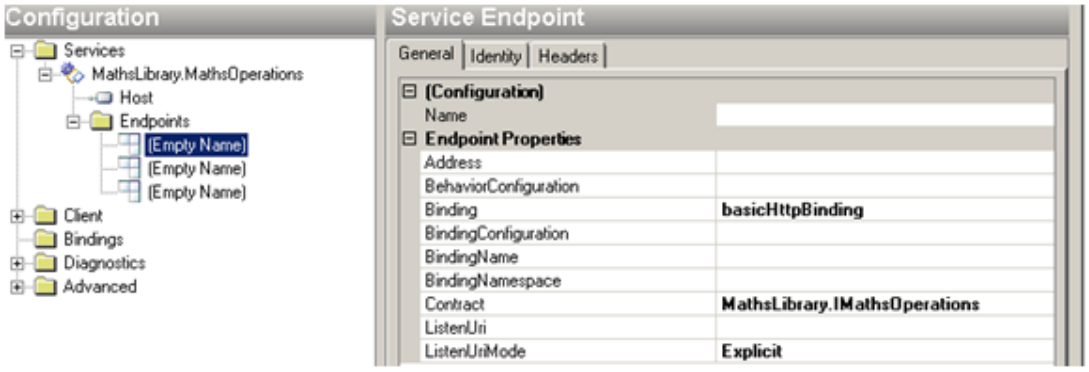
Cliquez sur le bouton "points de suspension" et trouvez MathsLibrary.dll.

Cliquez sur ce lien, il vous donnera le nom du service qu'elle contient.

De même aller sur des endpoints, et sélectionnez un contrat approprié avec les mêmes étapes que ci-dessus.

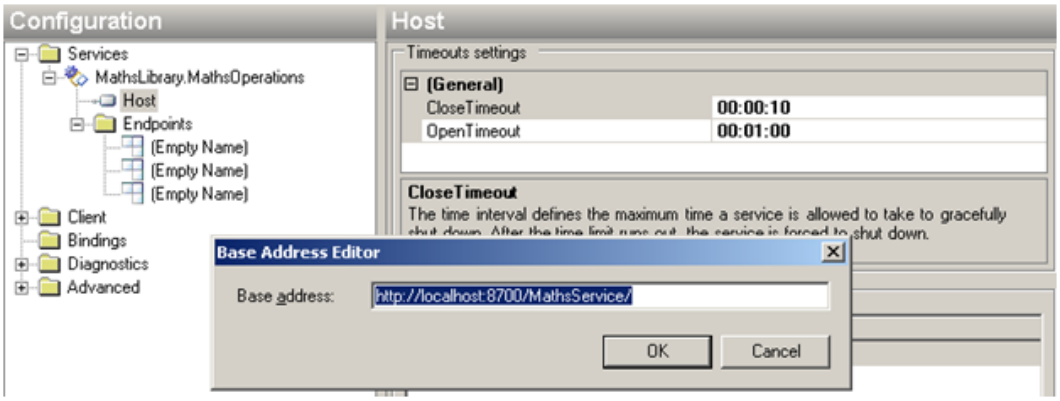


Nous pouvons aussi changer le binding comme ci-dessous …



Sélectionnez "Host" sur le côté gauche et vous verrez l'adresse de base. Vous pouvez la modifier vers l'adresse que vous voulez.

Applications Réparties
Services WCF (Windows Communication Foundation)



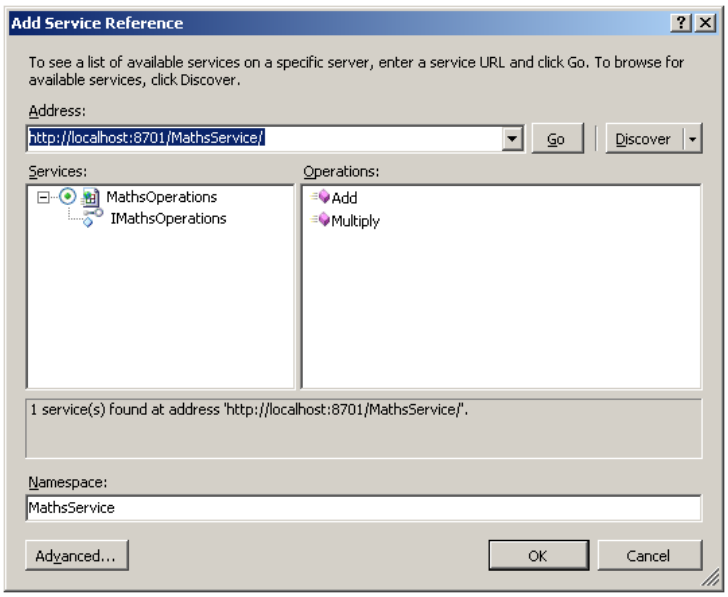
Une fois que cela est fait, générez le projet.

Visual Studio founit son propre prcocessus d'accueil d'hébergement de services. Nous pouvons également héberger le service comme une application console ou un Service Windows, ou encore dans IIS.

2 Création d'un client WCF avec C

Maintenant, nous allons créer une application cliente. Ajouter un projet "application console" (ou tout autre type de projet) dans la même solution ou une solution différente.

Ajouter une référence au service MathsService que nous avons créé. Il chargera automatiquement toutes les DLL nécessaires.



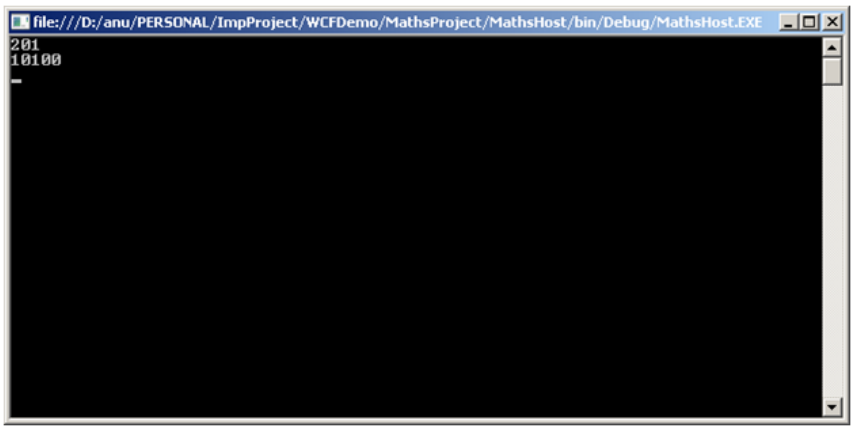
Puis, avec le code ci-dessous, nous pouvons appeler le service :

Applications Réparties
Services WCF (Windows Communication Foundation)

```
public class Program
{
    public static void Main(string[] args)
    {
        MathsOperationsClient client = new MathsOperationsClient();
        Console.WriteLine(client.Add(100,101));
        Console.WriteLine(client.Multiply(100, 101));

        Console.ReadLine();
    }
}
```

Cela vous donnera dessous de la sortie



3

4 Exploration de différents Bindings

Sans modifier vos codes sources, modifiez et testez de nouveaux bindings entre le client et le serveur.

Créer pour cela plusieurs endpoints avec des bindings différents et testez-les.

Attention dans ce cas de passer le nom du endpoint que vous voulez adresser lors d'une instanciation de la classe proxy sur le service.

Exemple MathsOperationsClient client = new MathsOperationsClient("BasicHTTP");

Dans le cas :

```
<endpoint address="" binding="basicHttpBinding" name="BasicHTTP",
contract="MathsLibrary.IMathsOperations">
      <identity>
            <dns value="localhost" />
      </identity>
</endpoint>
```