

Introduction à ASP.NET

Partie 2 (traitement des collections de données)

Auteurs : Ph. Lacomme (placomme@isima.fr) et R. Phan (phan@isima.fr)

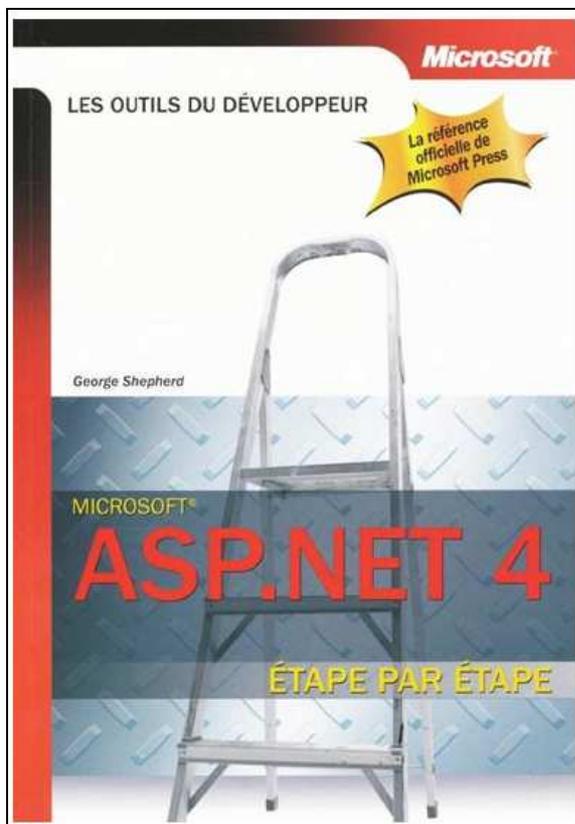
Date : juin 2011

Avertissement : Les exemples proposés dans ce tutorial viennent en partie du livre de George Shepherd. Nous avons refait certains exemples du livre, nous les avons complétés et nous espérons avoir amélioré les explications (parfois très rapides) qui sont données dans le livre.



Ceci n'est pas un cours.

Nous vous conseillons le livre suivant :



George Shepherd
Editeur : Microsoft Press, 2010 - 606 pages
ISBN : 2100547429

1. Contenu

Les collections de données se divisent en deux catégories :

- les collections sans liaisons de données ;
- les collections avec liaisons de données.

2. Liaison de données simple

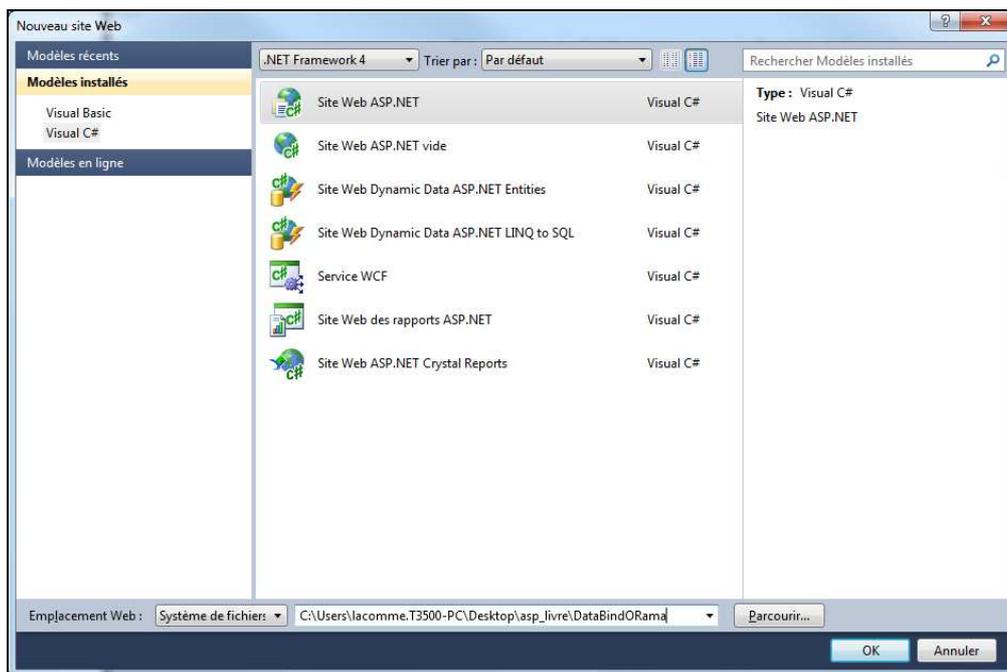


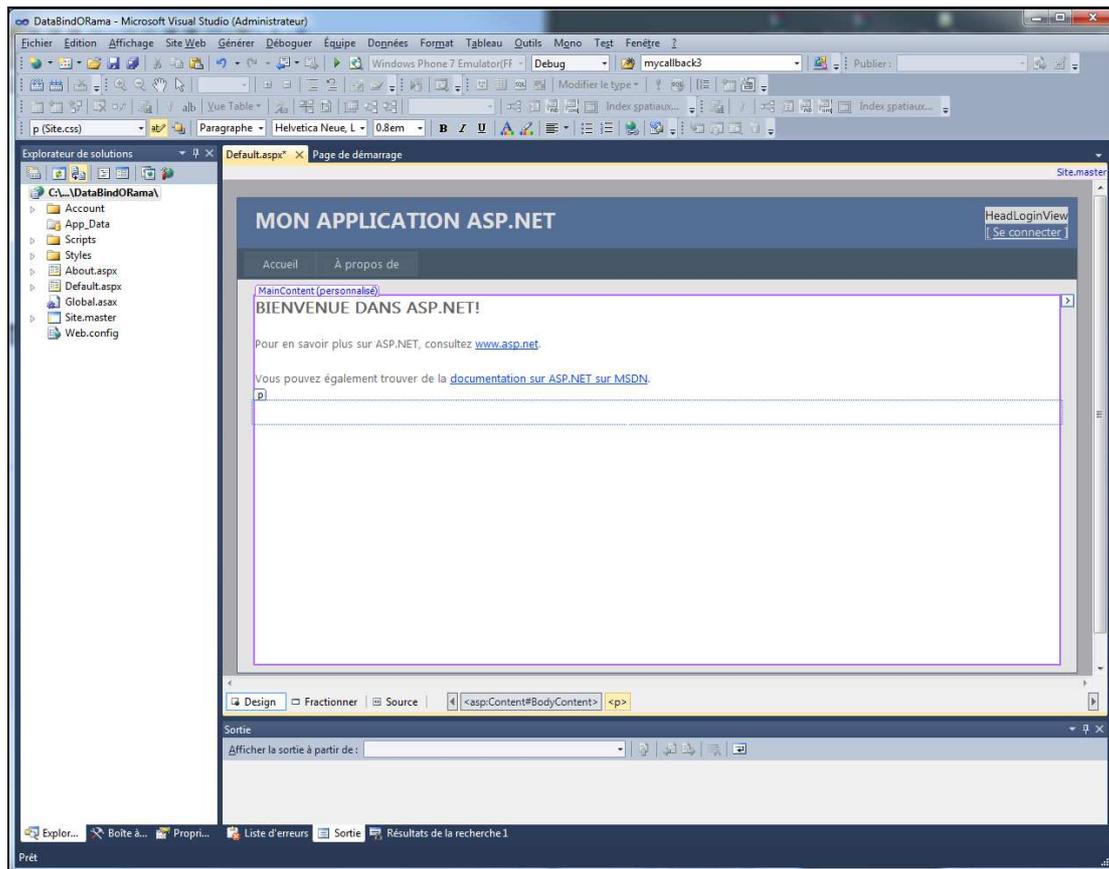
Objectifs :

- comprendre comment générer des données simplement ;
- comprendre comment les stocker ;
- comprendre comment les visualiser.

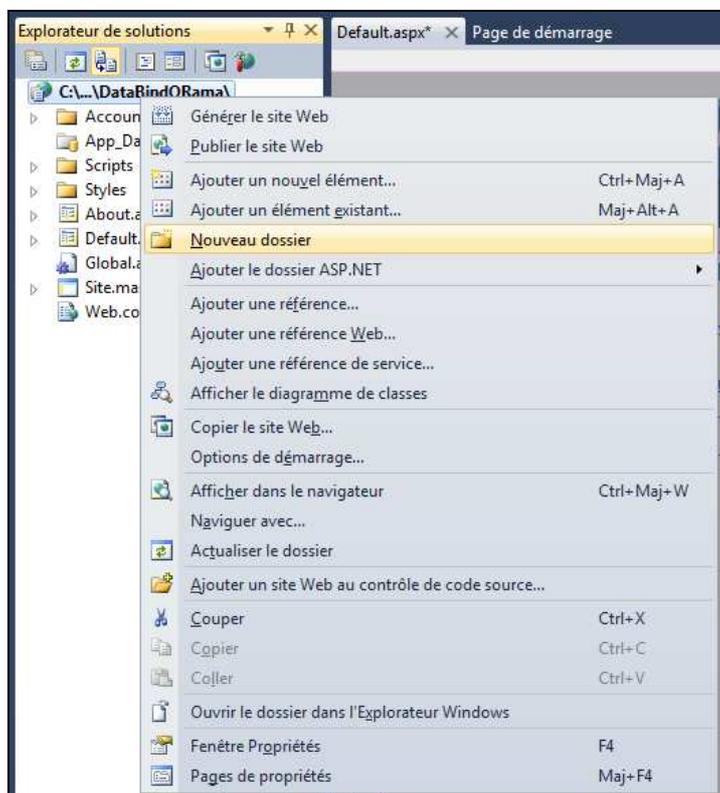
On va attacher une collection simple à la propriété DataSource de l'un des contrôles basés sur ListControl.

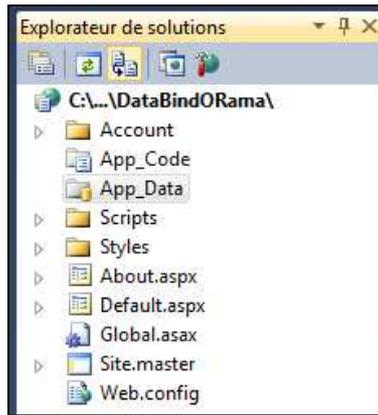
2.1. Créer un site web vide nommé DataBindORama.





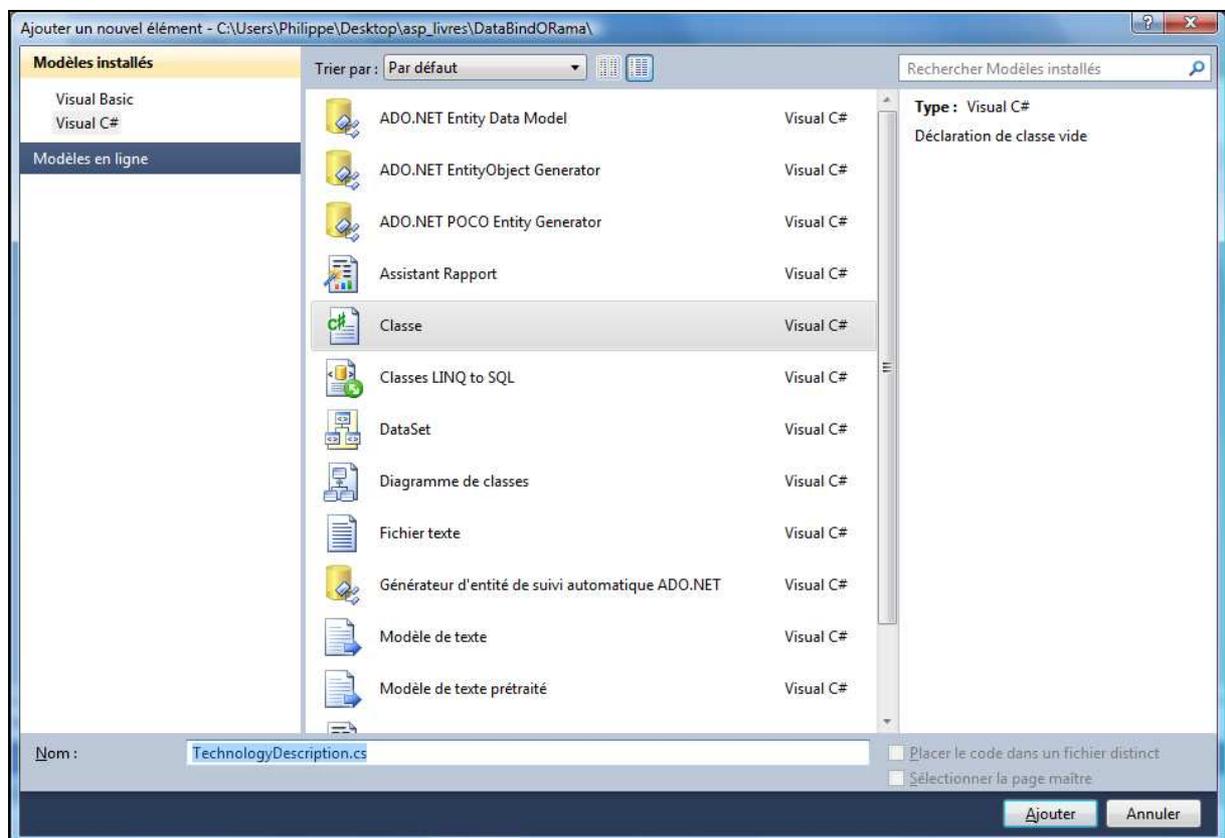
Ajouter un répertoire **App_Code**.





Faire un click droit sur le répertoire App_Code et choisir **Ajouter un nouvel élément**.

Ajouter une **classe** et nommé **TechnologyDescriptor.cs**



Nous allons ajouter les setters and getters dans cette classe de la manière suivante :

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

public class TechnologyDescriptor
{
    //
    // TODO: ajoutez ici la logique du constructeur

    public string TechnologyName { get; set; }

    public string Description { get; set; }

    public TechnologyDescriptor (string strTechnologyName,
        string strDescription)
    {
        this.TechnologyName = strTechnologyName;
        this.Description = strTechnologyName;
    }

    public static List<TechnologyDescriptor> CreateTechnologyList()
    {
        List<TechnologyDescriptor> iTechnologies = new List<TechnologyDescriptor>();
        TechnologyDescriptor maTechnologieDescriptor;
        maTechnologieDescriptor = new TechnologyDescriptor("ASP.NET", "traiter une requette
HTTP");
        iTechnologies.Add(maTechnologieDescriptor);

        maTechnologieDescriptor = new TechnologyDescriptor("Windows Form", "Technologie");
        iTechnologies.Add(maTechnologieDescriptor);

        maTechnologieDescriptor = new TechnologyDescriptor("ASP.NET et Linq", "Communique avec
la base de donnees");
        iTechnologies.Add(maTechnologieDescriptor);

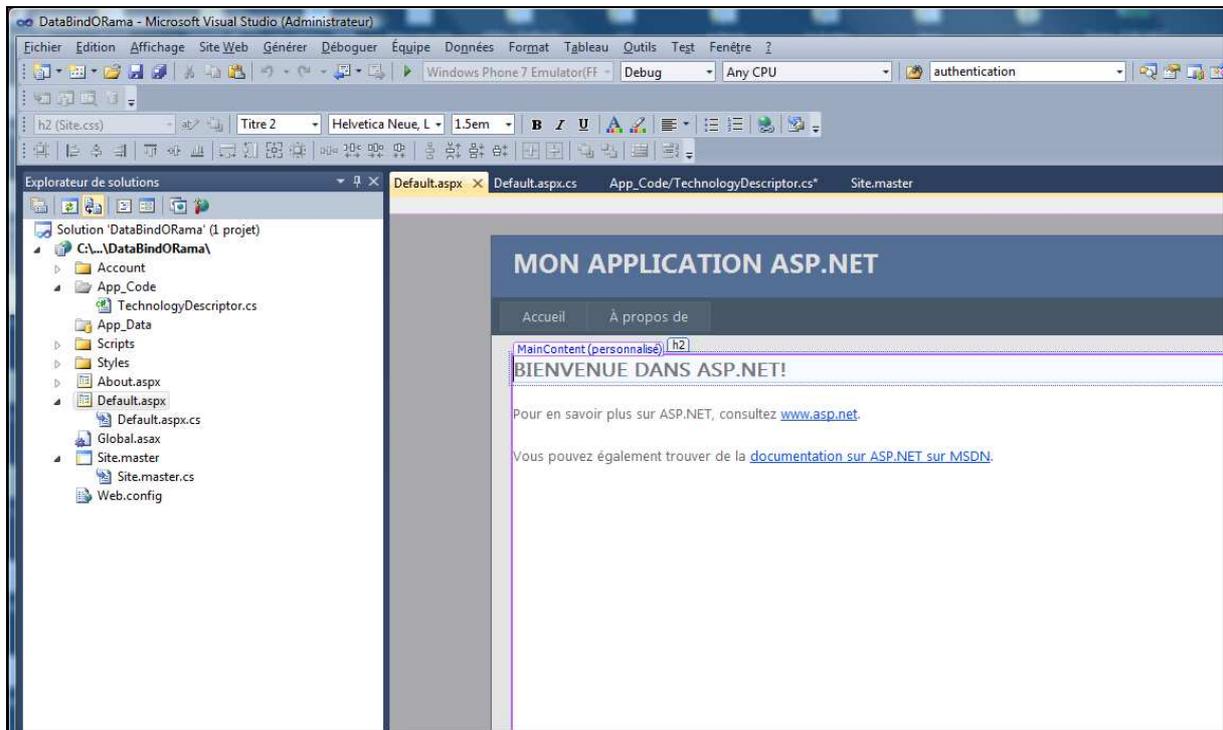
        maTechnologieDescriptor = new TechnologyDescriptor("ASP.NET env.", "Environnement
d'execution");
        iTechnologies.Add(maTechnologieDescriptor);

        maTechnologieDescriptor = new TechnologyDescriptor("WPF", "Technologie de rendu
avancee");
        iTechnologies.Add(maTechnologieDescriptor);

        return iTechnologies;
    }
}
```

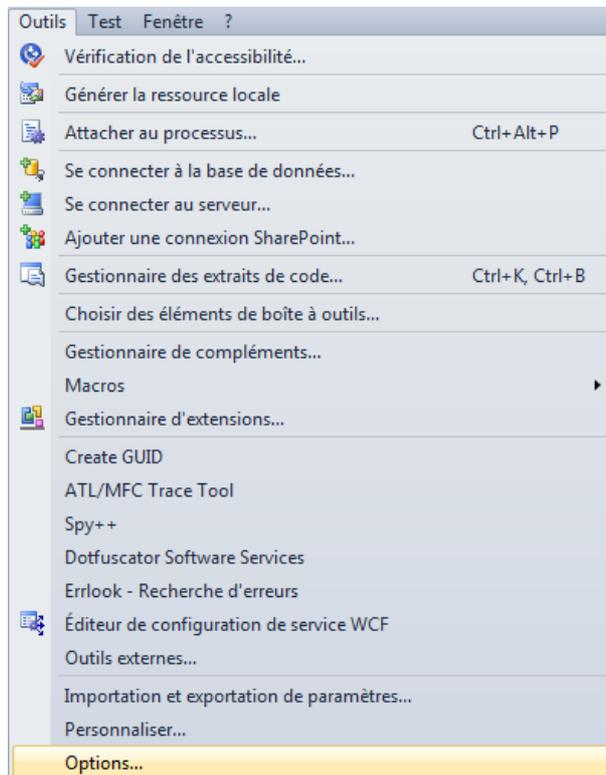
2.2. Modifiez le contenu de la page.

Ouvrir le fichier Default.aspx et basculez en mode design.

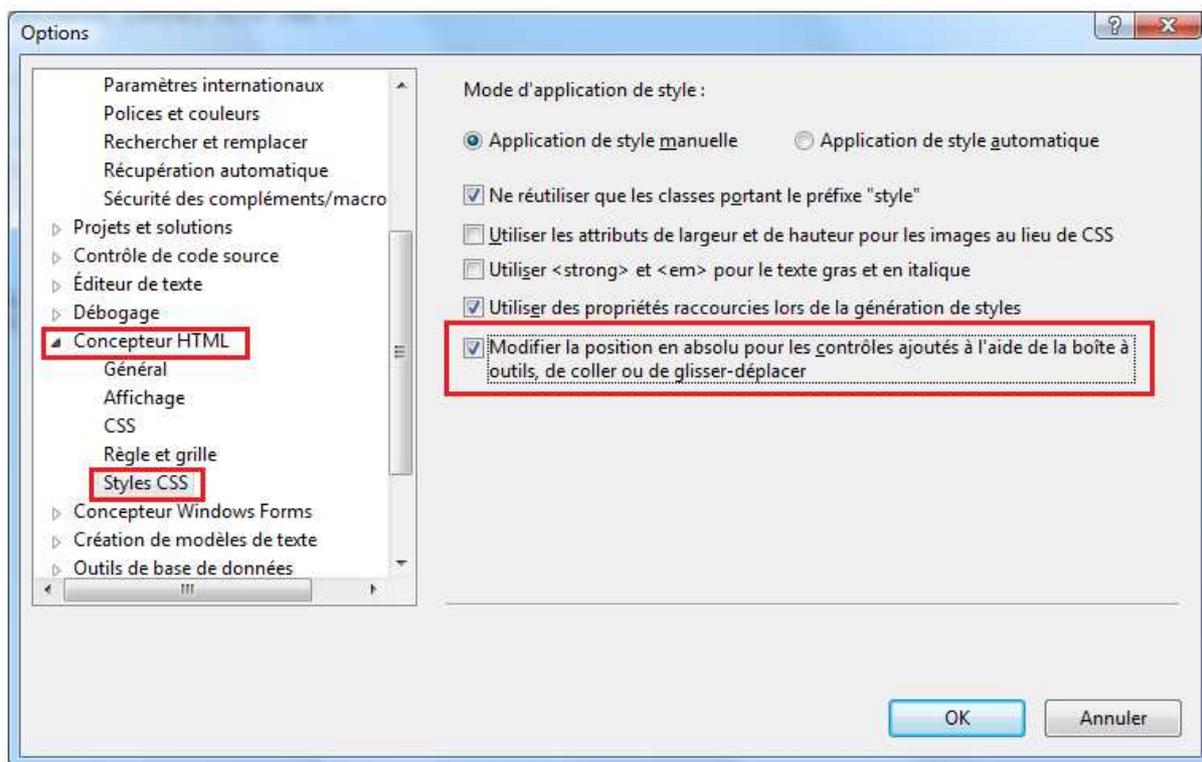


Basculez la gestion des éléments en mode de **positionnement absolu**.

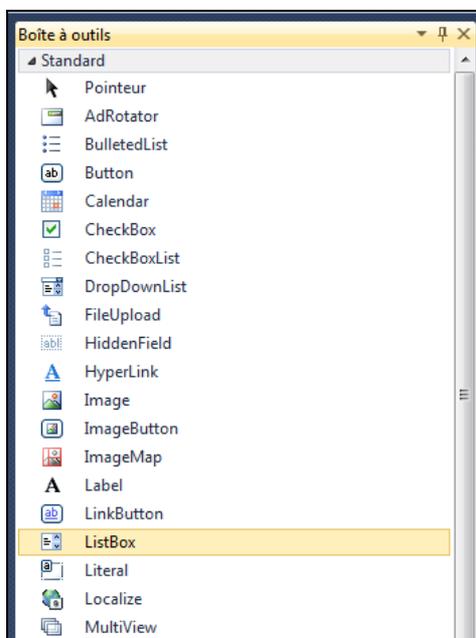
Allez dans dans **Outils /Options**.



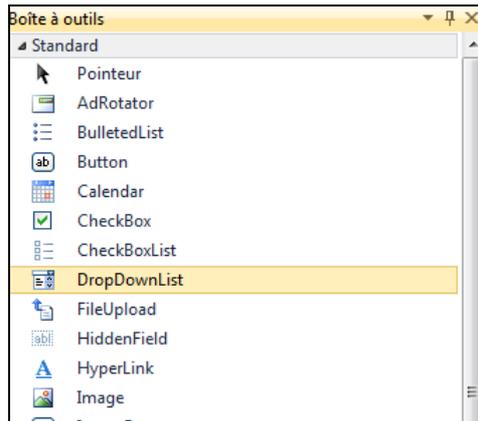
Choisir ensuite : **Concepteur HTML / Styles CSS.**



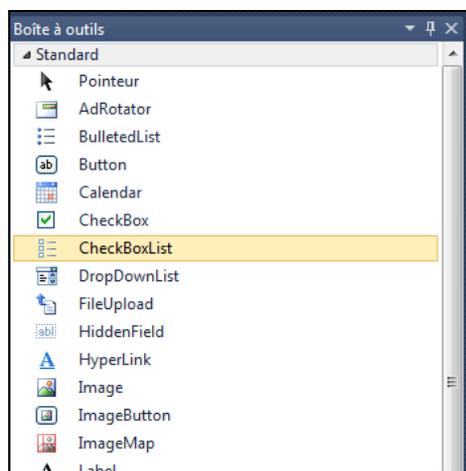
Insérez un **ListBox**.



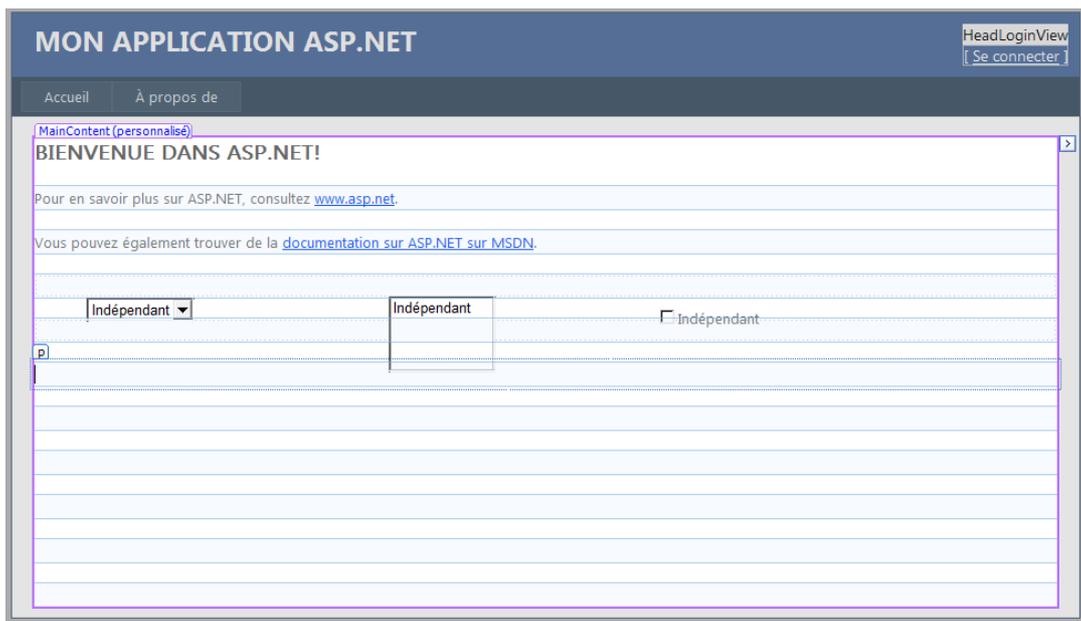
Insérer un élément **DropDownList**.



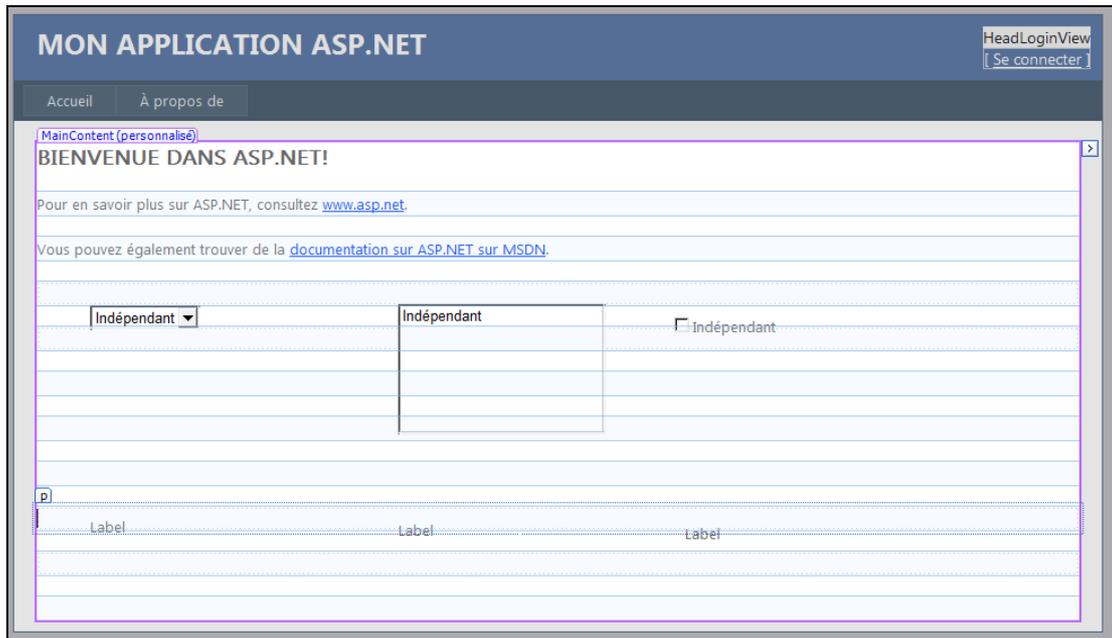
Insérer un élément **CheckBoxList**.



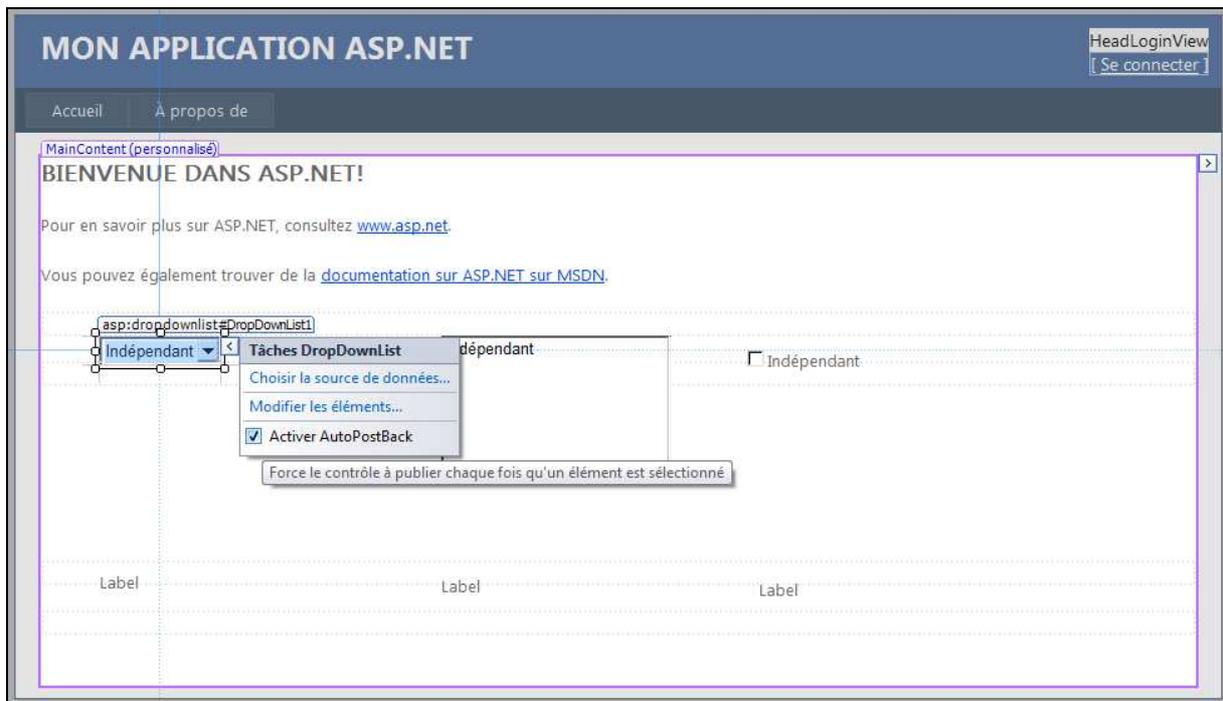
L'application devrait ressembler à ce qui suit :

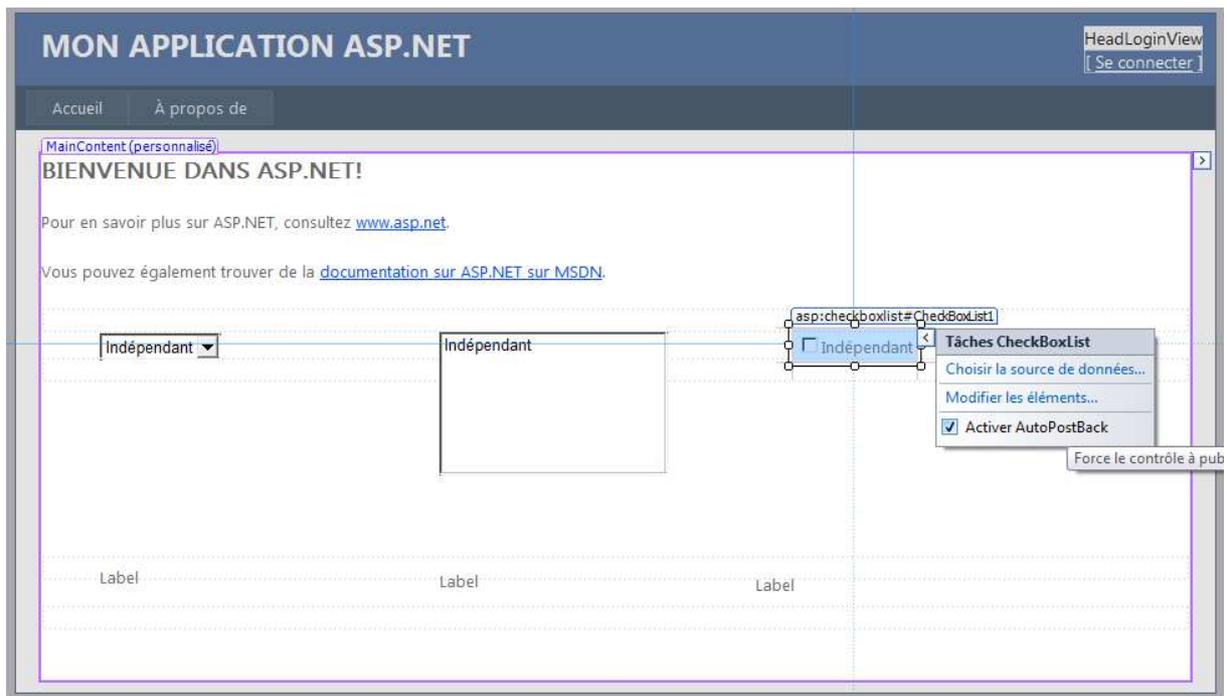
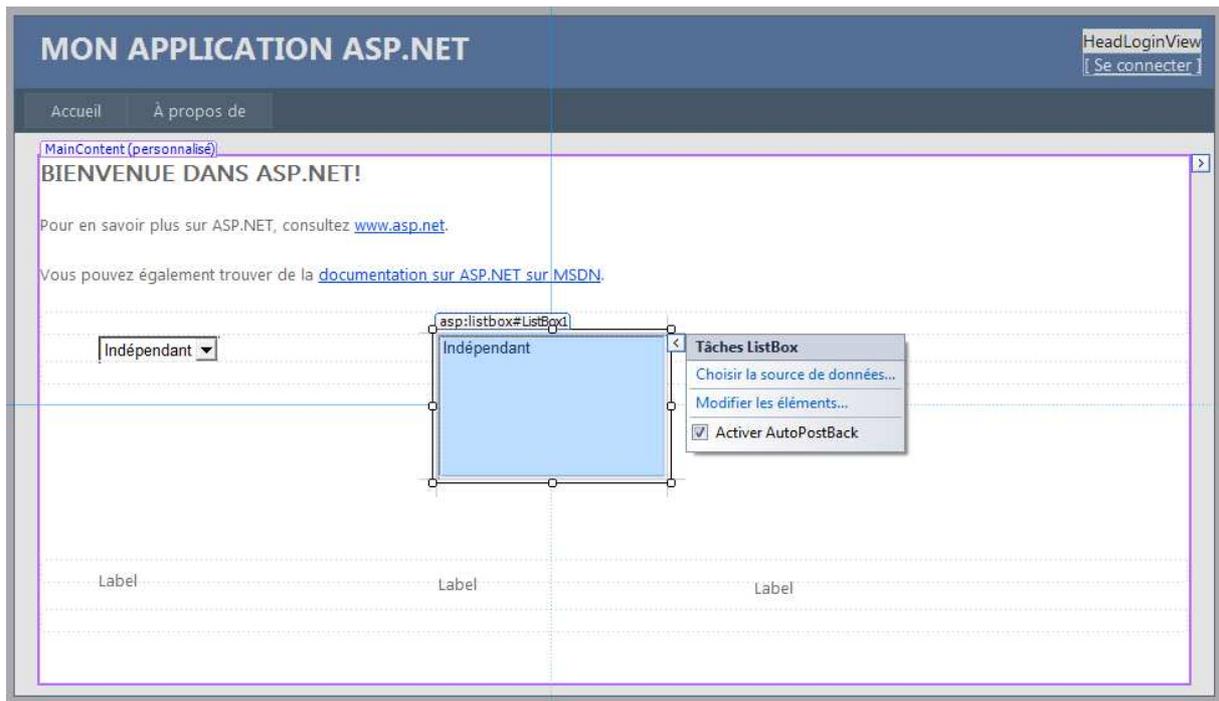


Rajoutez un label sous chacun des controles. Ils vont nous permettre d'afficher l'élément sélectionné par l'utilisateur.



Prendre chaque contrôle et cochez **Activez AutoPostBack**.





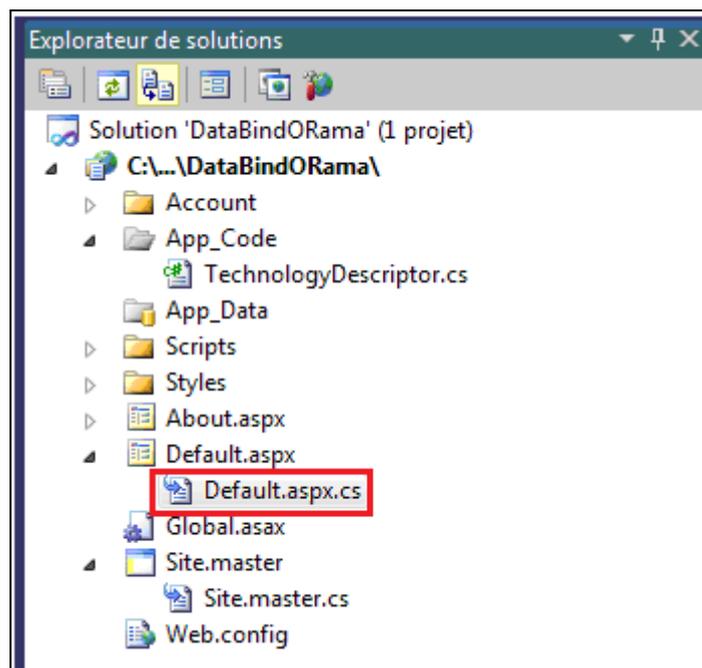
Pour chaque contrôle, modifier le champ **DataTextField**.

Données	
(Expressions)	
DataMember	
DataSourceID	
DataTextField	TechnologyName
DataTextFormatS	
DataValueField	

Modifiez ensuite le champ **DataValueField** de chaque élément comme ceci :

DataSourceID	
DataTextField	TechnologyName
DataTextFormat	
DataValueField	Description

Ouvrez le fichier **Default.aspx.cs**



Modifiez le code comme suit :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (this.IsPostBack == false)
        {
            List<TechnologyDescriptor> ma_liste = TechnologyDescriptor.CreateTechnologyList();

            this.ListBox1.DataSource = ma_liste;
            this.ListBox1.DataTextField = "TechnologyName";

            this.DropDownList1.DataSource = ma_liste;
            this.DropDownList1.DataTextField = "TechnologyName";

            this.CheckBoxList1.DataSource = ma_liste;
            this.DropDownList1.DataTextField = "TechnologyName";

            this.DataBind();
        }
    }
}
```

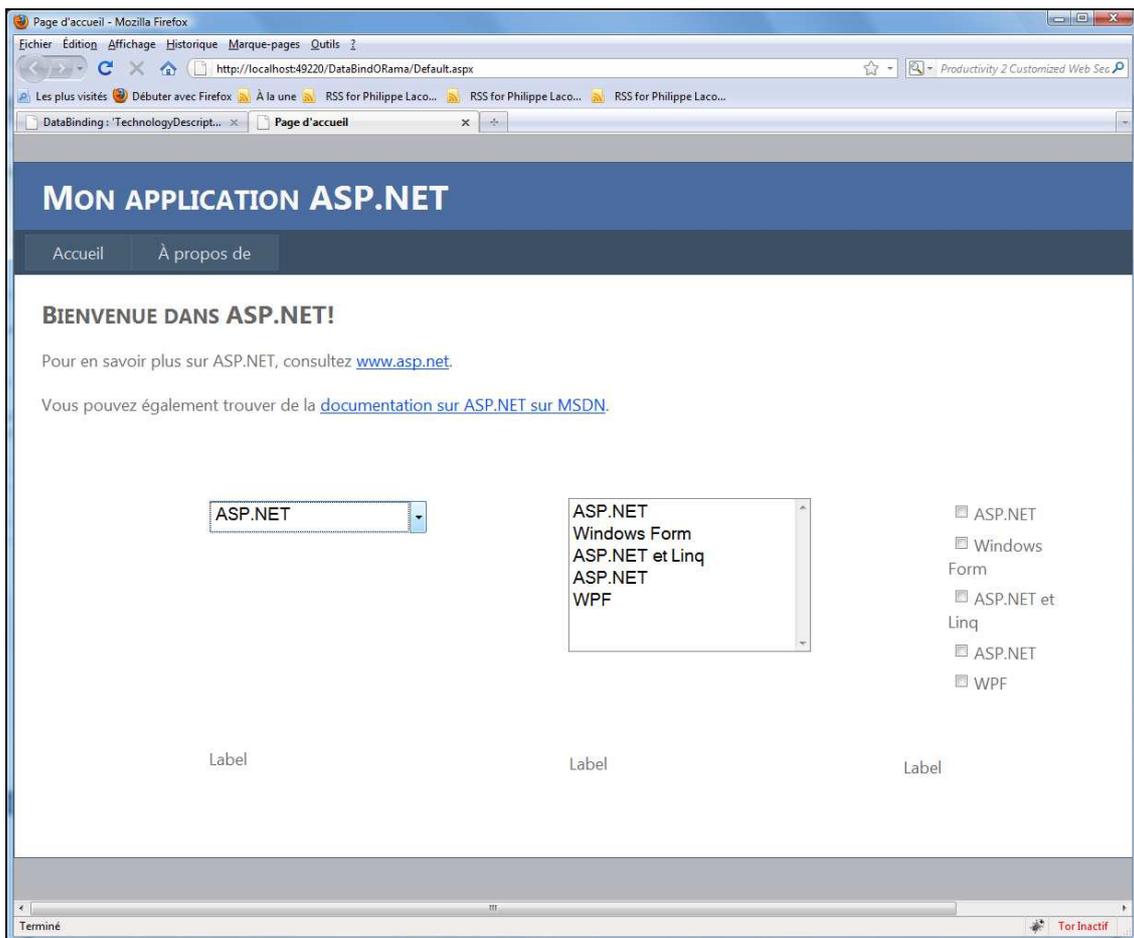
Attention

Ne pas oublier le : `if (this.IsPostBack == false)`...

Si cette condition n'est pas présente, tout rechargement de la page (et c'est la cas sur les évènements click) se traduira par un rechargement du contenu des objets. On aura l'impression que la sélection à la souris d'un élément ne fonctionne pas.

Remarquez que le champ DataTextField fait référence à TechnologyName que nous avons précédemment saisie.

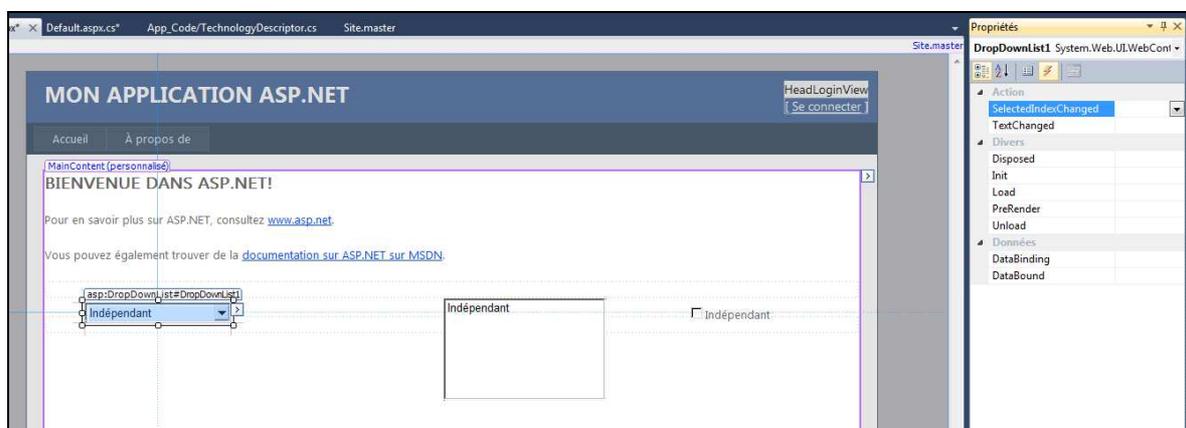
Le résultat d'exécution est le suivant :



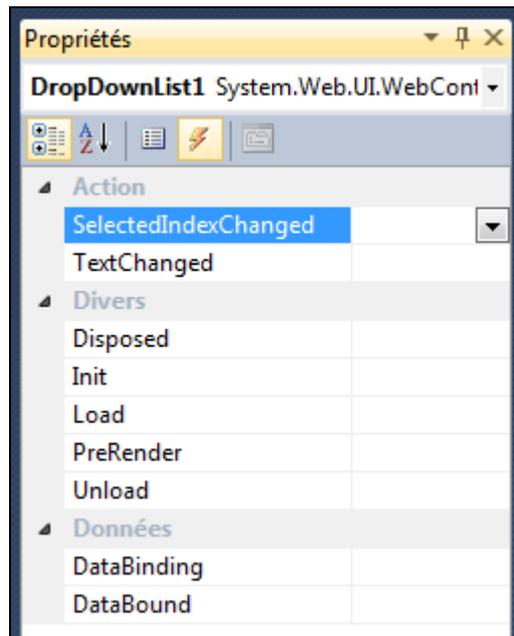
2.3. Récupérer les éléments choisis par l'utilisateur.

Editez le fichier **Default.aspx**.

Selectionnez **ListBox1** et consultez les évènements.



Choisir dans la liste **SelectedIndexChanged**.



Faire un double-click sur le champ.

Visual Studio génère alors une méthode comme celle-ci :

```
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
}
```

Modifiez la méthode comme ceci :

```
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
    string chaine = this.DropDownList1.SelectedValue;
    this.Label1.Text = chaine;
}
```

Réitérez l'opération avec les trois éléments.

Le code final doit ressembler à ceci :

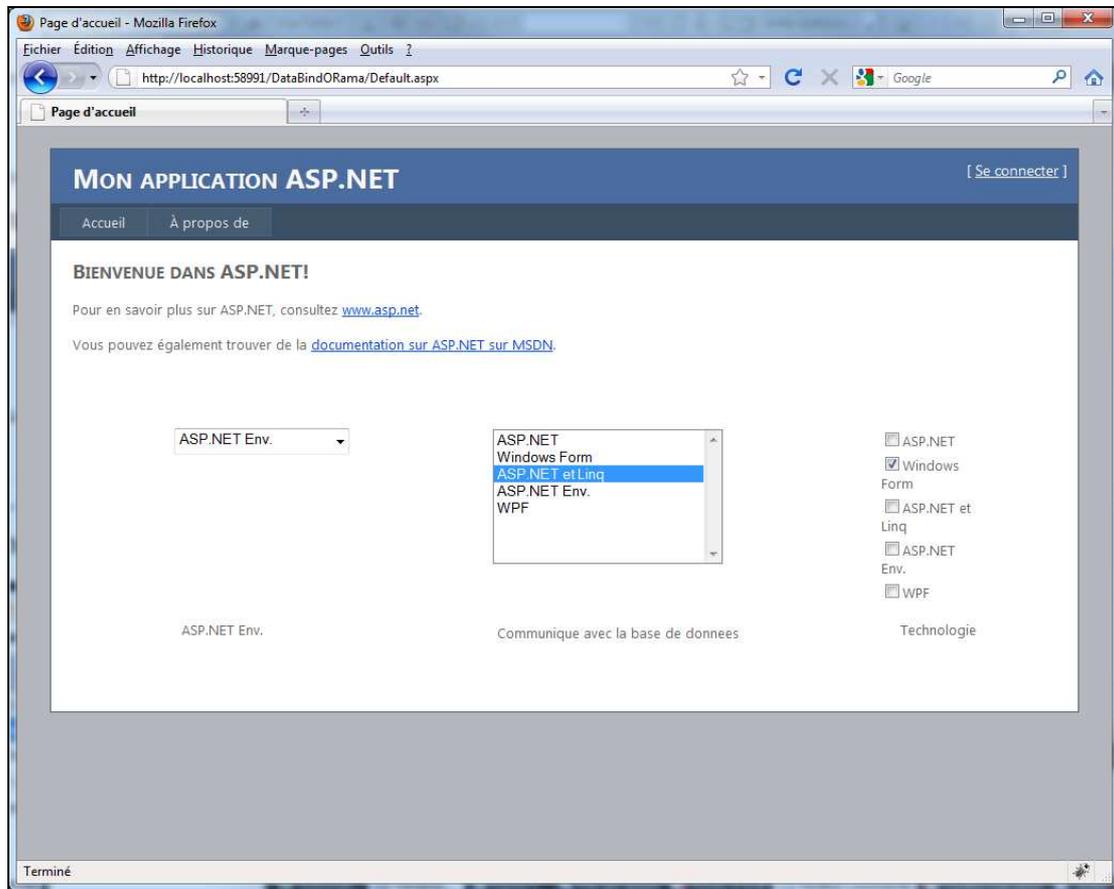
```
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
    string chaine = this.DropDownList1.SelectedValue;
    this.Label1.Text = chaine;
}

protected void ListBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    string chaine = this.ListBox1.SelectedValue;
    this.Label2.Text = chaine;
}

protected void CheckBoxList1_SelectedIndexChanged(object sender, EventArgs e)
{
    string chaine = this.CheckBoxList1.SelectedValue;
    this.Label3.Text = chaine;
}
```

Lancez l'application.

On peut constater que le fait de sélectionner un élément entraîne automatiquement l'affichage de l'élément sélectionné.



2.4. Conclusion

Nous venons de montrer comment attacher des ensembles d'éléments en mémoire (que nous avons stocké dans des ArrayList et des List) à un contrôle. Nous avons aussi montré comment procéder afin de récupérer les sélections d'un utilisateur.

3. Accéder à une base de données

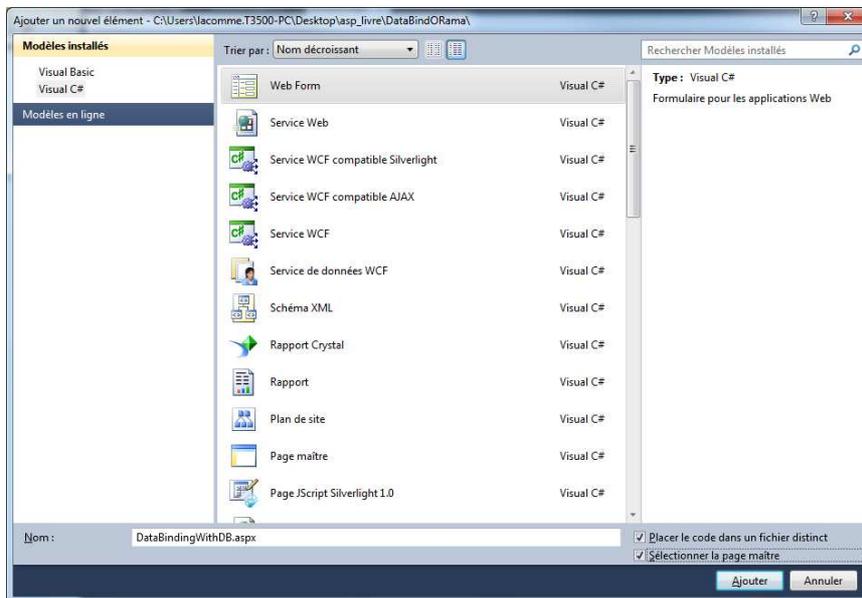


Objectifs :

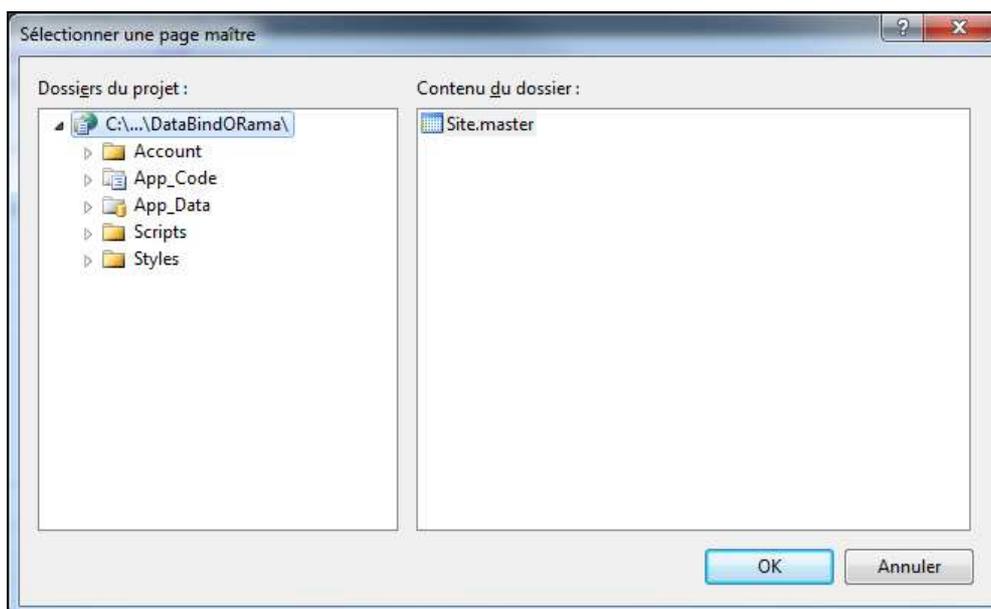
- Gérer une base de données via SQL Server.

3.1. Ajout d'un nouveau formulaire à DataBindORama.

Ajouter un nouveau formulaire nommé **DataBindingWithDB** .

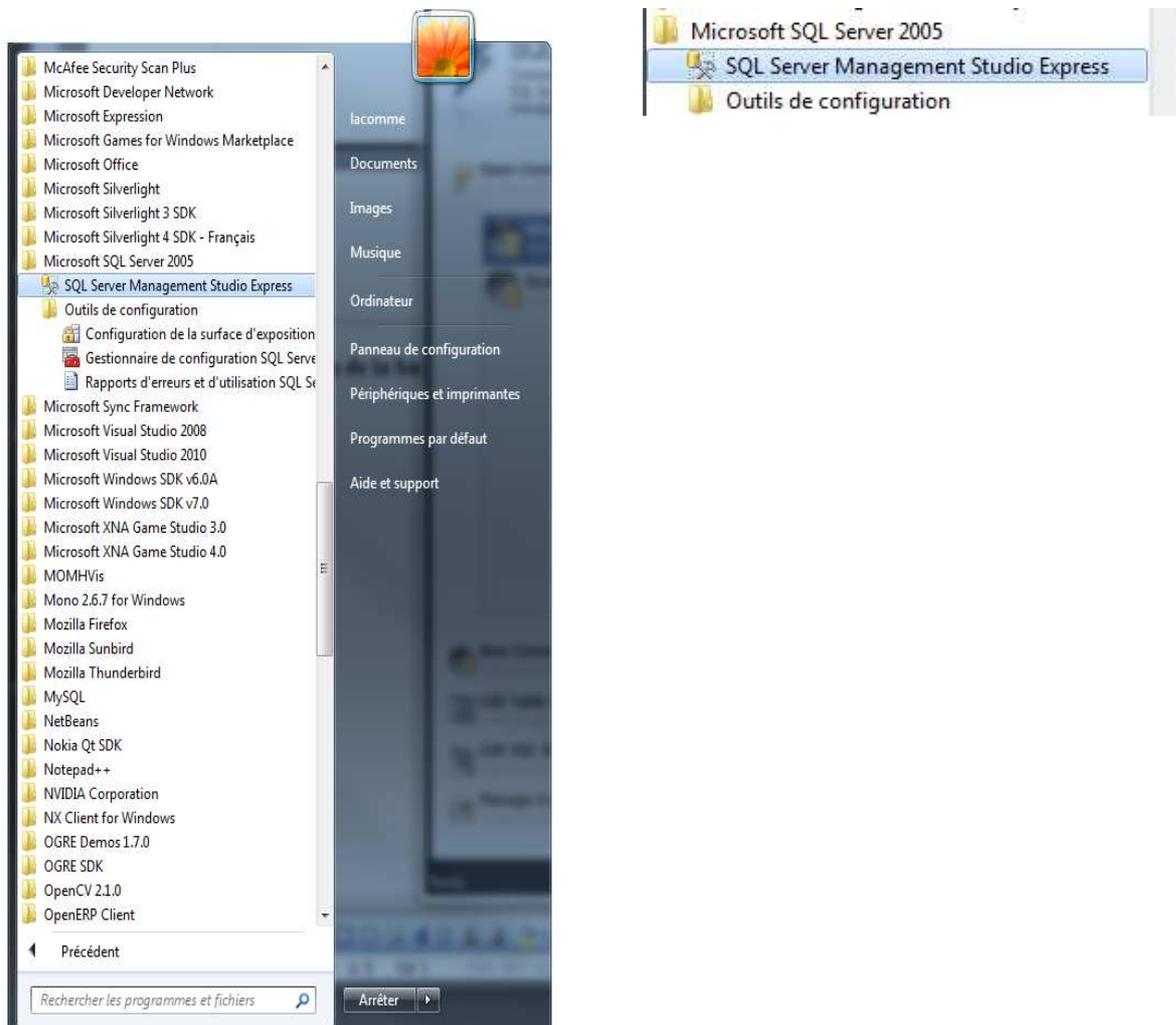


Choisissez comme page maître **Site.master**.



3.2. Création de la base de données

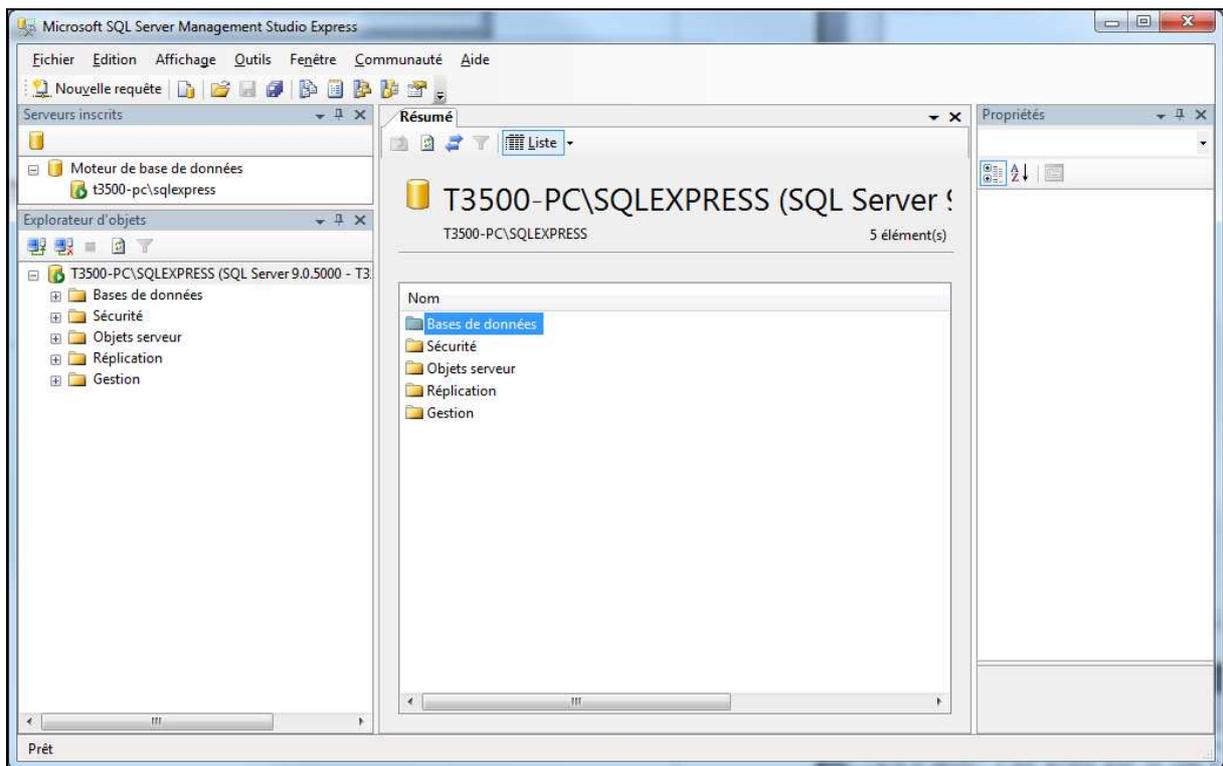
Allez dans le menu **Démarrer** et choisir **Microsoft SQL Server**.



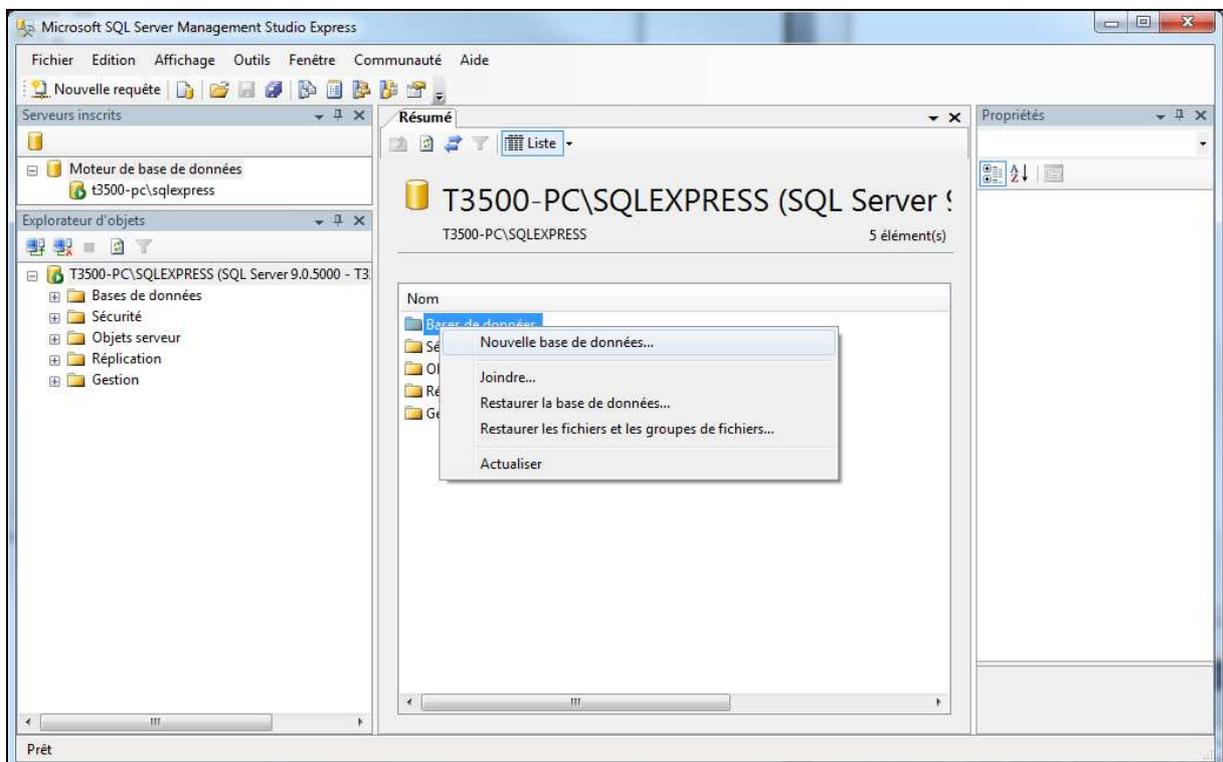
Acceptez la connexion.



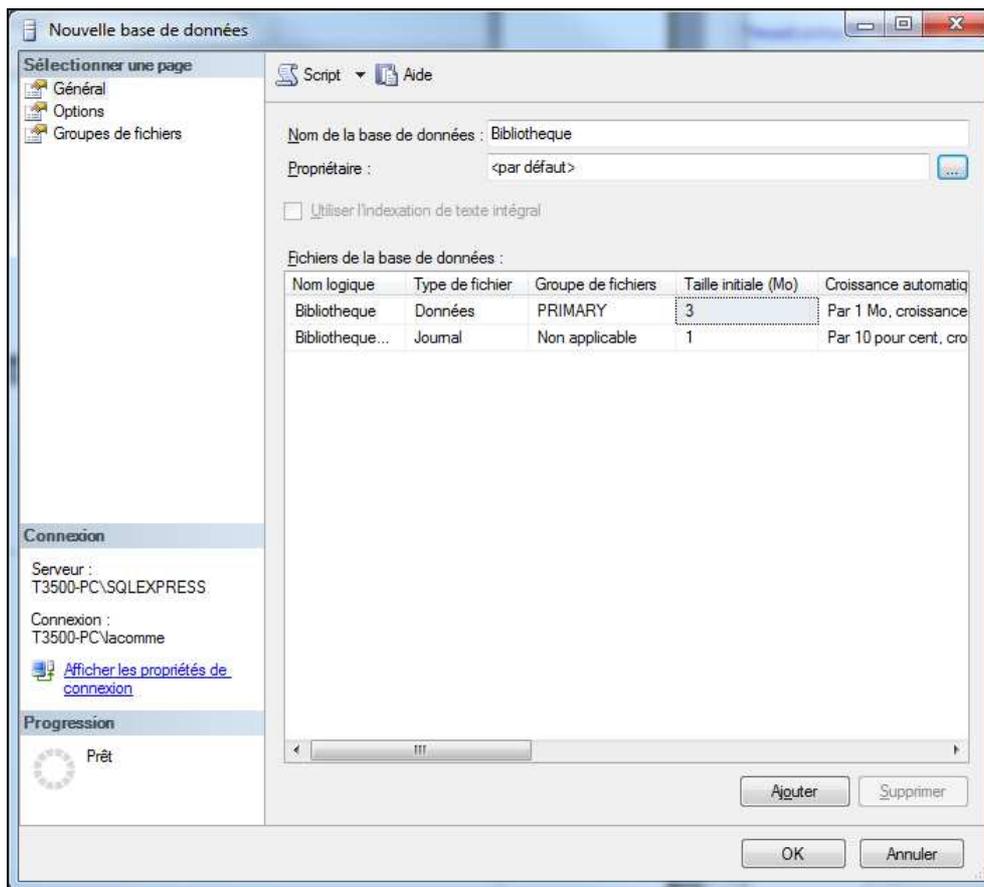
Le gestionnaire de base de données se présente comme suit :



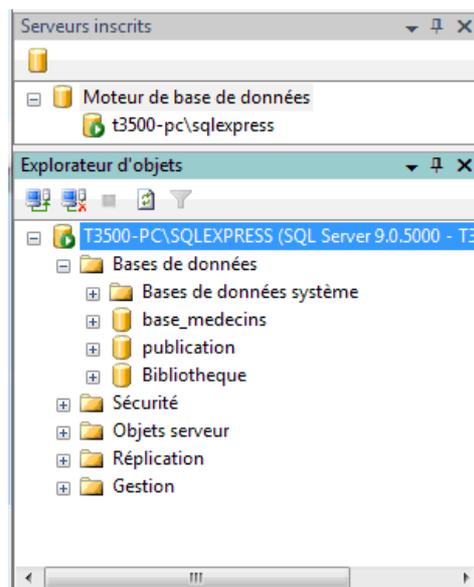
Faire un click droit sur **Base de données** au centre de la fenêtre et choisir **Nouvelle base de données**.



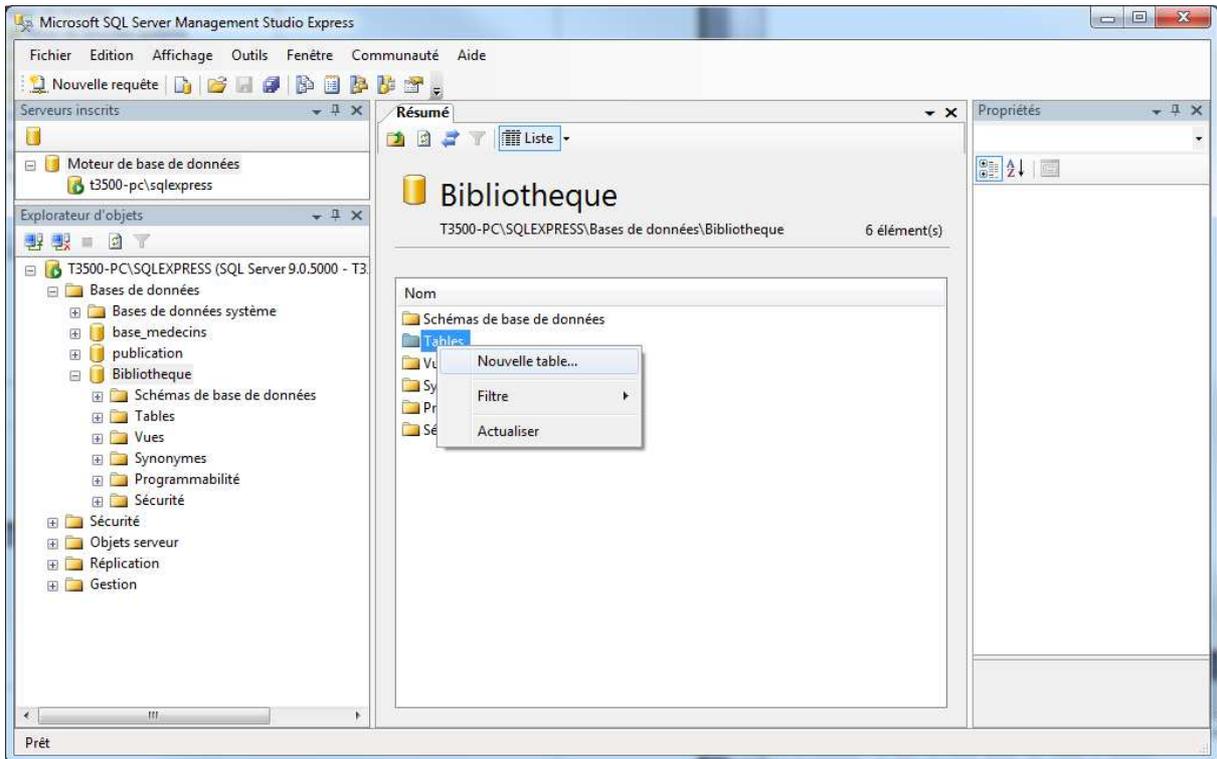
Choisir comme nom de la base de données : **Bibliothèque**.



Une fois créée, la base de données doit apparaître dans la liste



Il faut créer une nouvelle table. Pour cela sélectionner la base **Bibliothèque** et faire **Nouvelle Table**.



Créer une table nommée **Auteur** ressemblant à ce qui suit :

Nom de la colonne	Type de données	Null autorisé
Nom	nchar(10)	<input type="checkbox"/>
Prenom	nchar(10)	<input type="checkbox"/>
Age	nchar(10)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

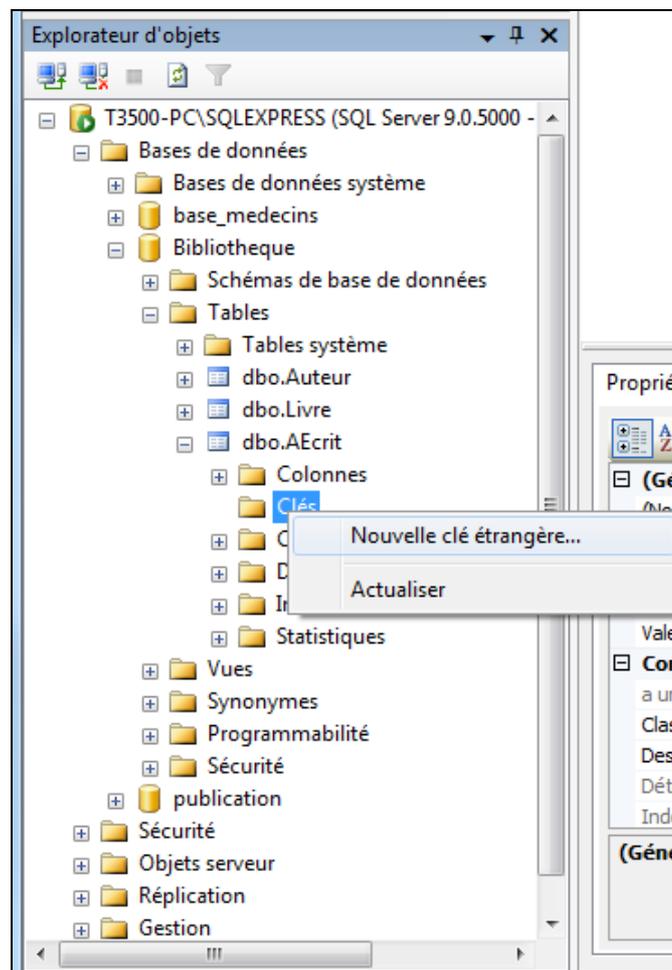
Créer une table nommée **Livre** ressemblant à ce qui suit :

Nom de la colonne	Type de données	Null autorisé
Titre	nchar(10)	<input type="checkbox"/>
prix	nchar(10)	<input type="checkbox"/>
ISBN	nchar(10)	<input type="checkbox"/>
		<input type="checkbox"/>

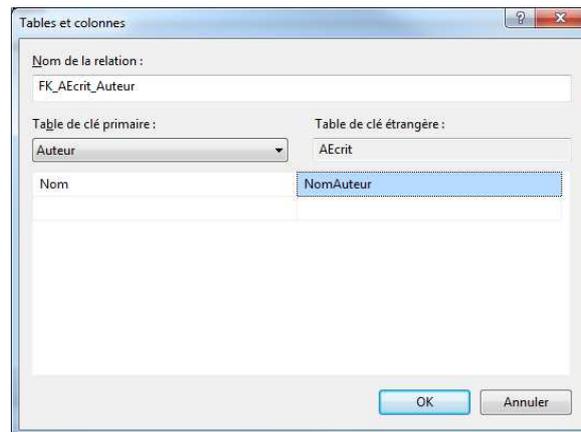
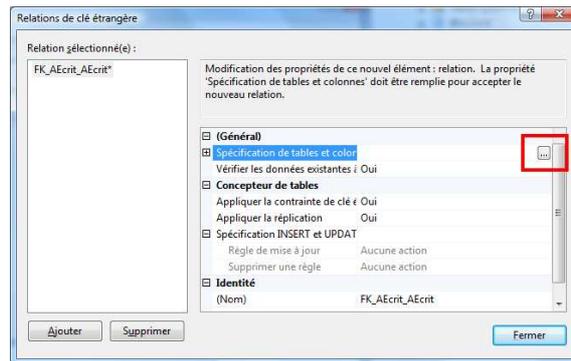
Créer une table nommée **AEcrit** ressemblant à ce qui suit :

Nom de la colonne	Type de données	Null autorisé
NomAuteur	nchar(10)	<input type="checkbox"/>
ISBNLivre	nchar(10)	<input type="checkbox"/>
		<input type="checkbox"/>

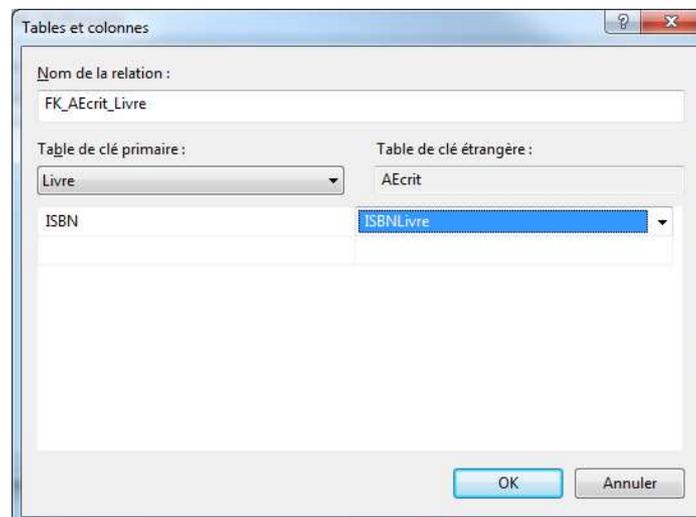
Dans l'explorateur d'objets, choisir la table **AEcrit** et dans la section **Clés**, choisir **Nouvelle clé étrangère**.



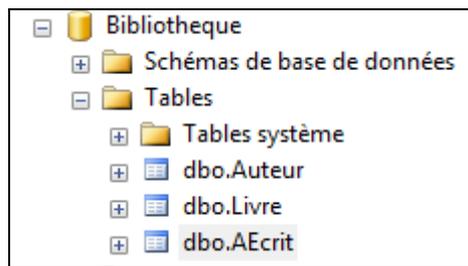
Mettre en relation la clé **Nom** de la table **Auteur** et la clé **NomAuteur** de la table **AEcrit**.



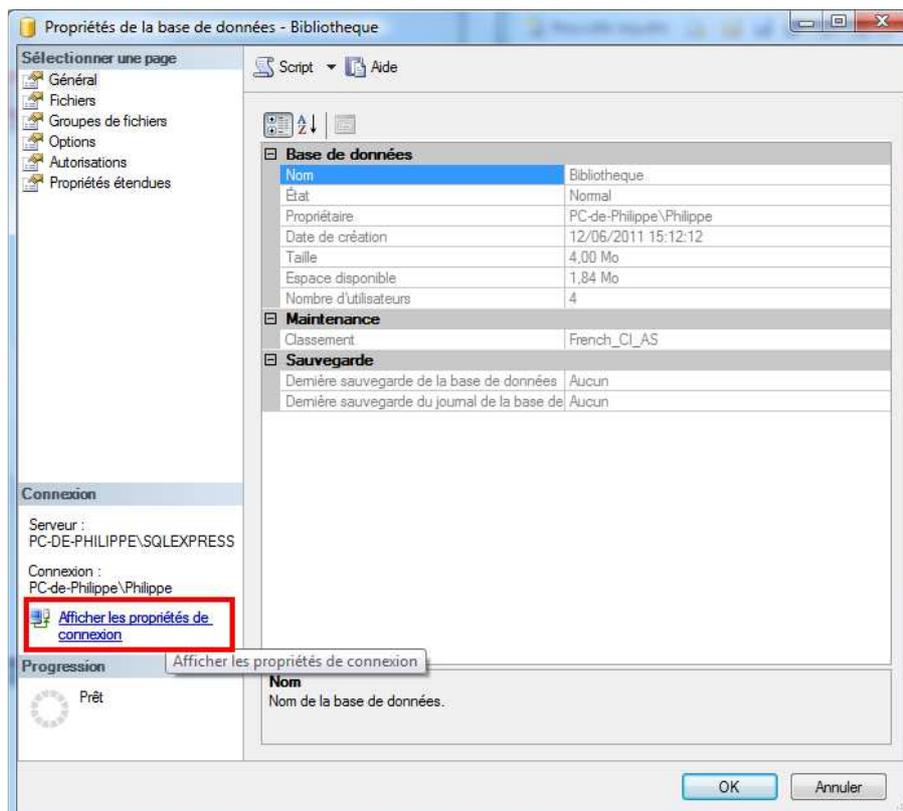
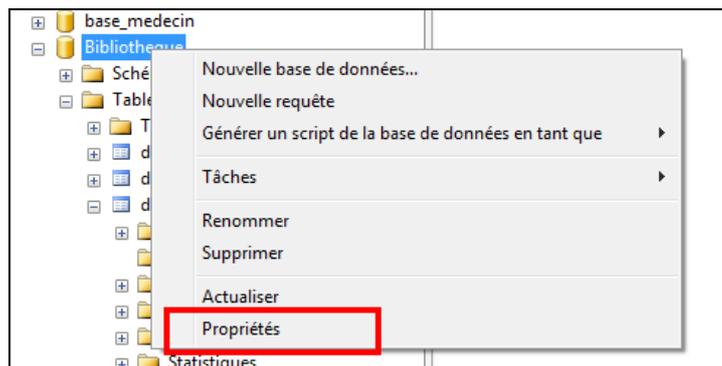
Mettre en relation la clé **ISBN** de la table **Livre** et la clé **ISBN** de la table **AEcrit**.

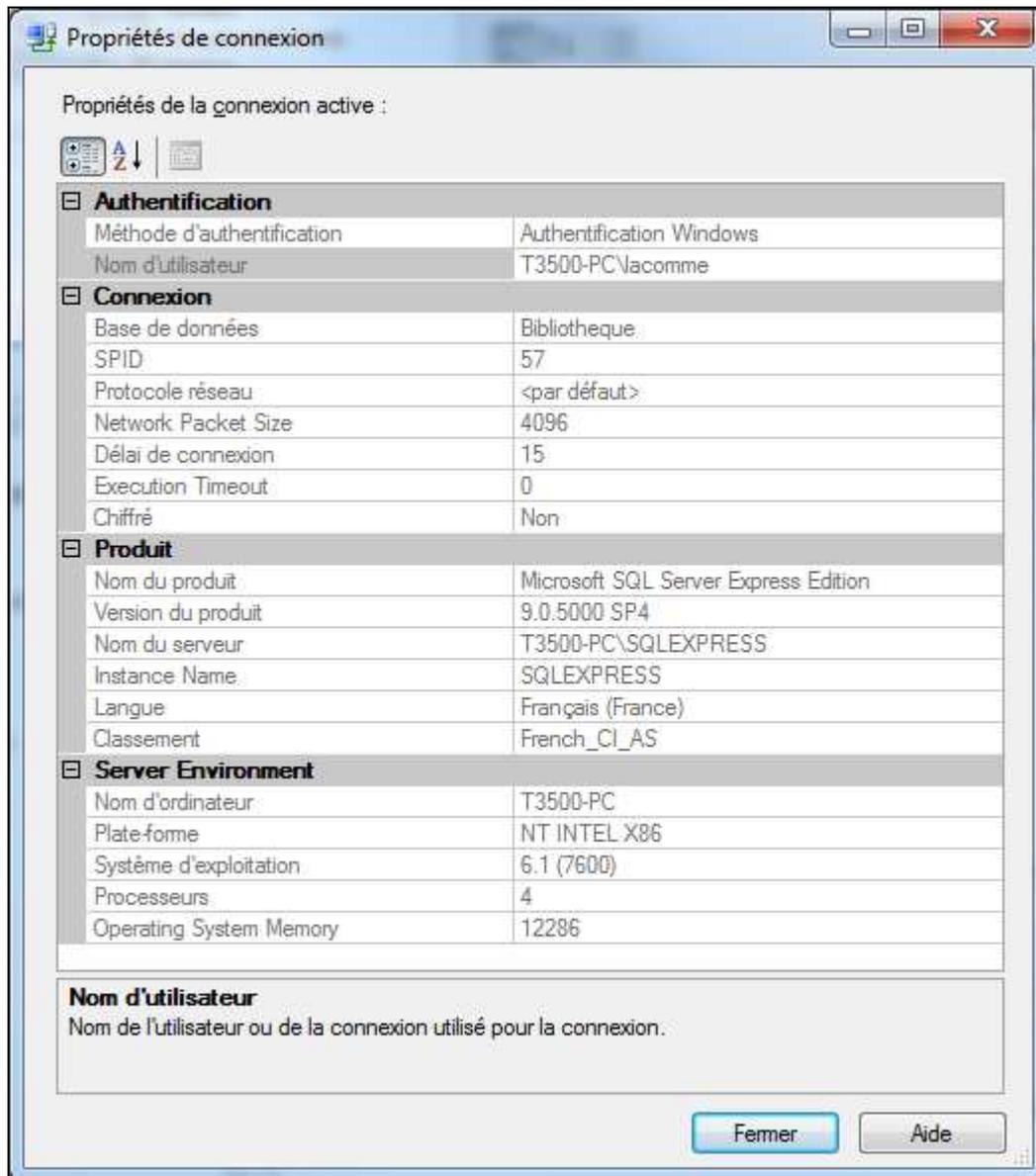


La base de données se présente comme suit :



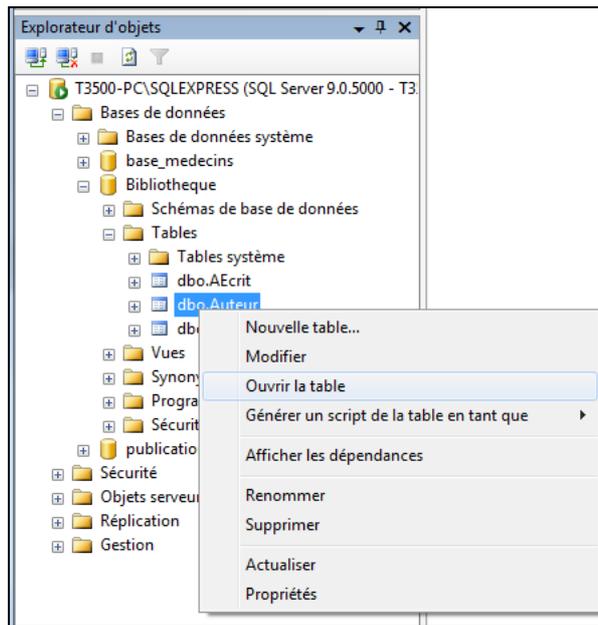
Visualisez les propriétés de connexion :





Il est important ici de noter le nom du serveur.
Sur ma machine : **T3500-PC\SQLEXPRESS**.

Faire un click droit sur **Auteur** et choisir **Ouvrir la table**.



La table est alors vide.

Table - dbo.Auteur			
	Nom	Prenom	Age
*	NULL	NULL	NULL

Remplir la table avec par exemple des informations similaires à celles-ci :

Table - dbo.Auteur			
	Nom	Prenom	Age
	Castafiore	Emilie	78
	Dupont	Thierry	78
	Tintin	Pierre	45
	Milou	Mimi	42
▶*	NULL	NULL	NULL

Procédez de manière similaire avec la table Livre.

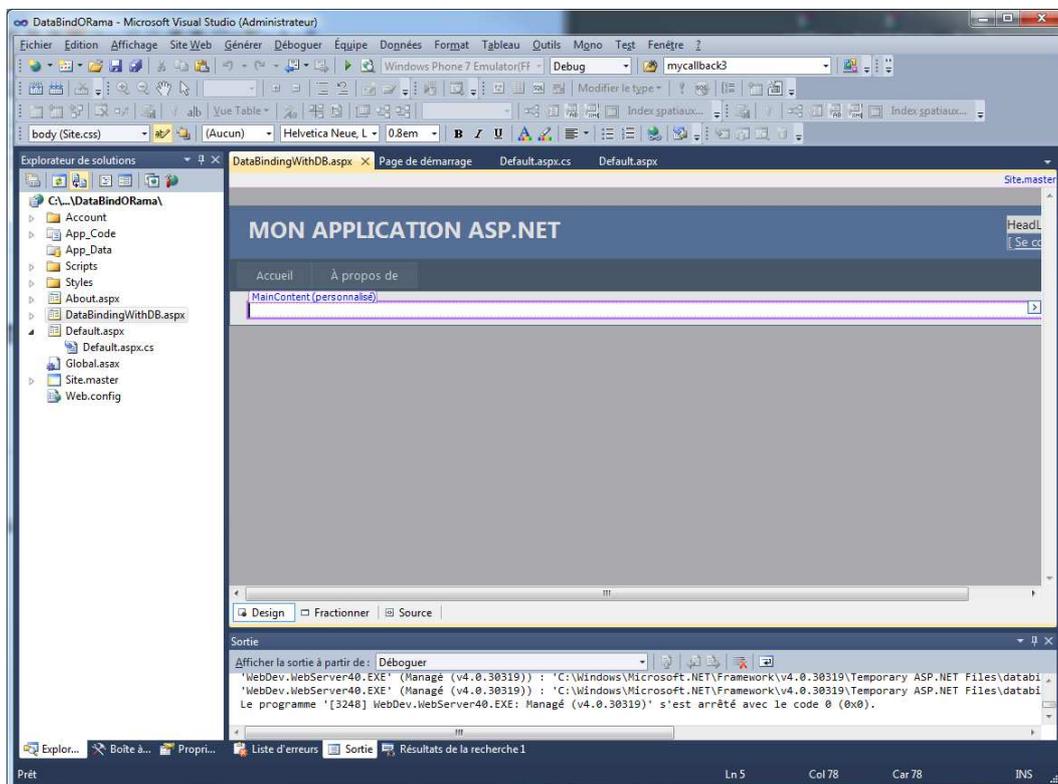
Table - dbo.Livre			
	Titre	prix	ISBN
	aaaaa	10	124587
	bbbb	25	9874
	cccc	85	960101
	dddd	101	010203
	eeee	202	852741
▶*	NULL	NULL	NULL

Terminez enfin par la table **AEcrit**.

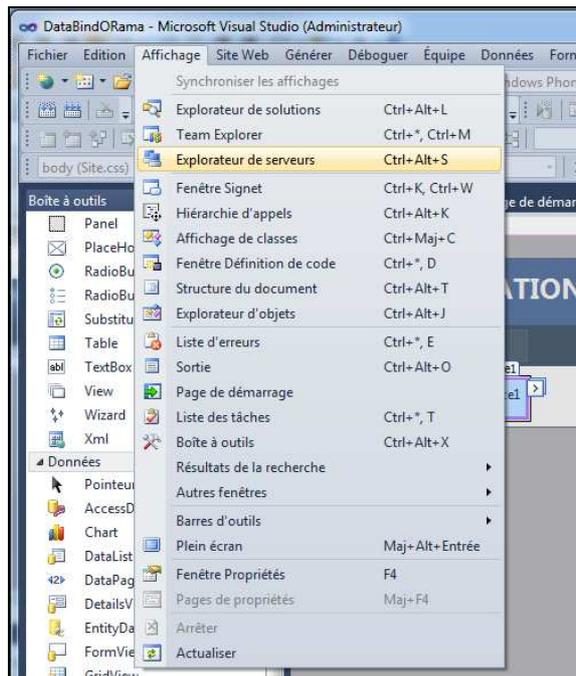
Table - dbo.AEcrit	Table - dbo.Livre	Table - dbo.Auteur	Résumé
	NomAuteur	ISBNLivre	
	Dupont	9874	
	Dupont	960101	
	Tintin	960101	
	Milou	010203	
▶*	NULL	NULL	

3.3. Connexion à la base de données

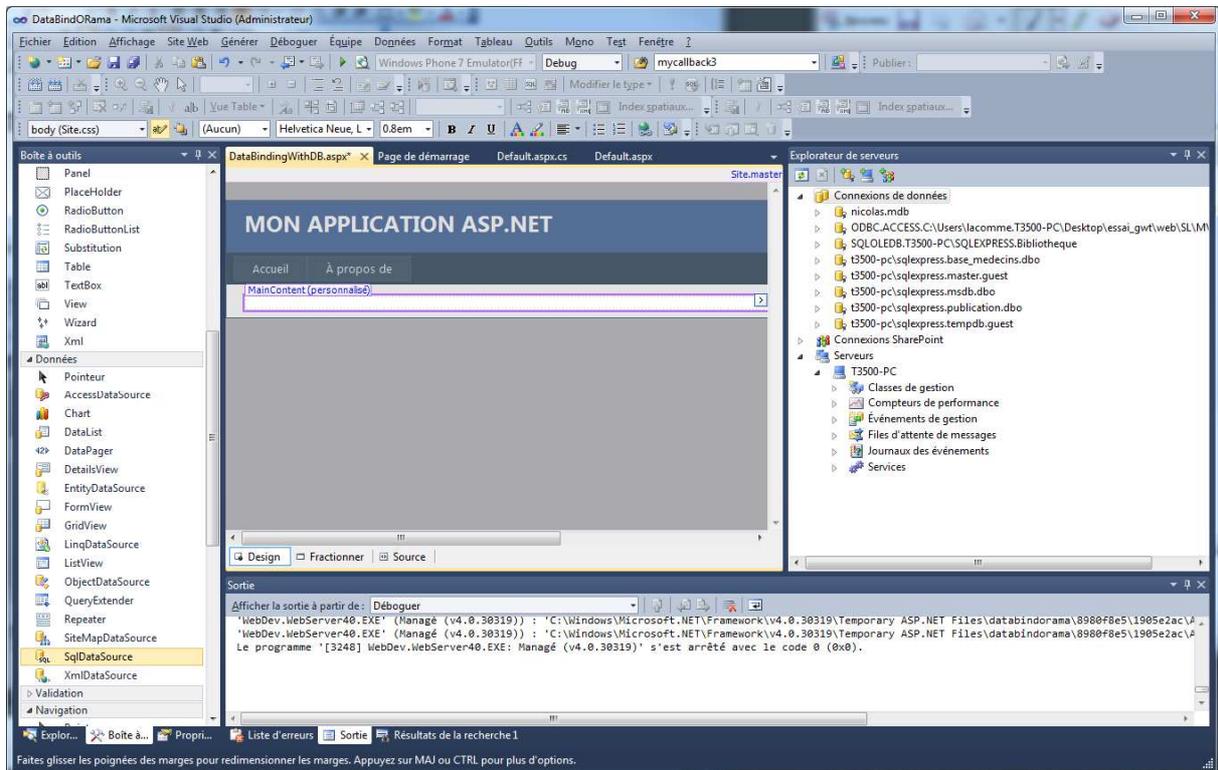
Revenez à l'environnement Visual Studio.
Visualisez la page **DataBindingWithDB.aspx**.



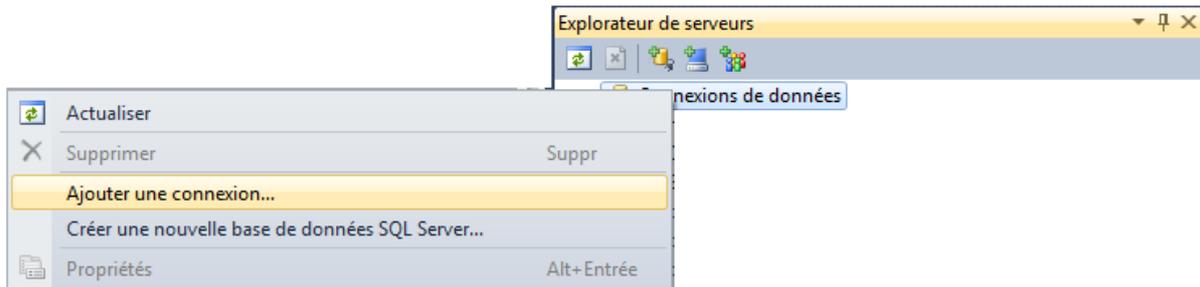
Allez dans le menu **Affichage** et choisir **Explorateur de serveurs**.



L'environnement se présente maintenant comme suit :

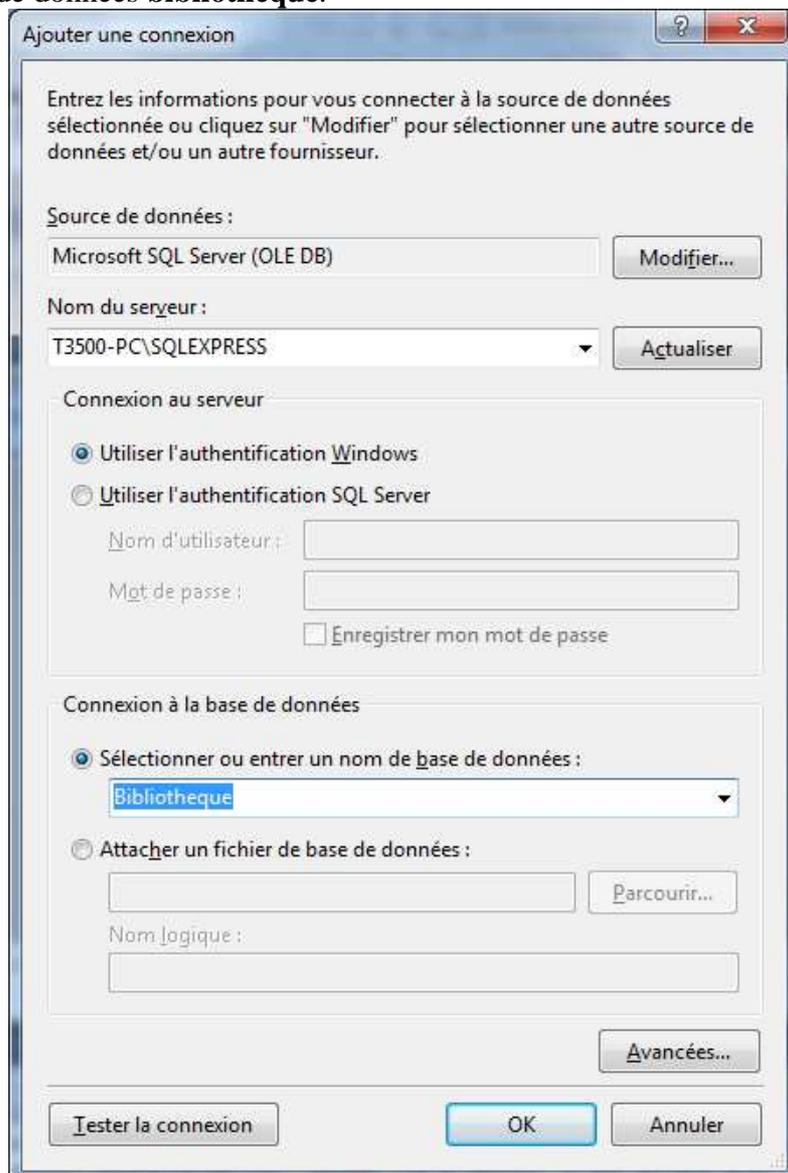


Dans l'explorateur de serveurs, choisir **Ajouter une connexion**.

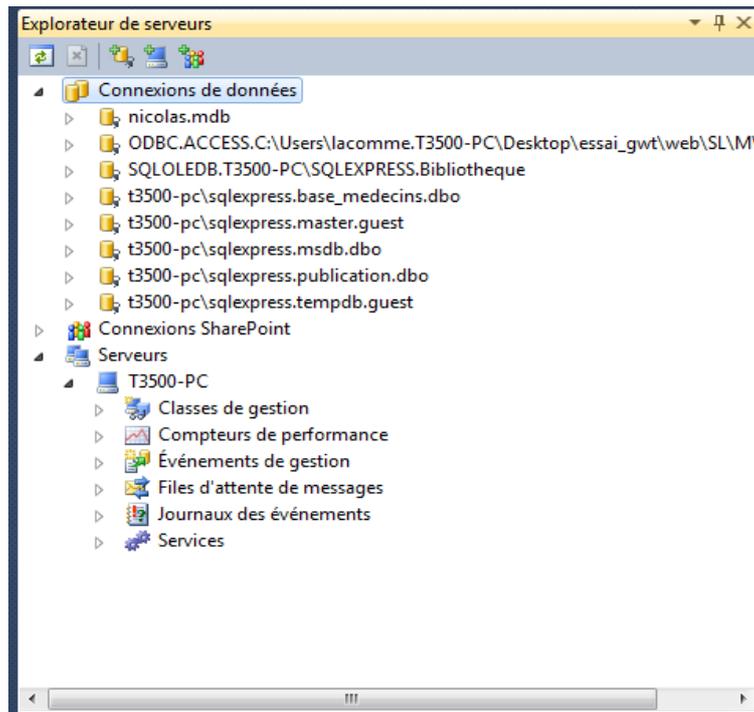


Ajouter comme nom de serveur : **T3500-PC\SQLEXPRESS**

La liste des bases de données enregistrées devient alors accessible.
Choisir la base de données **bibliotheque**.

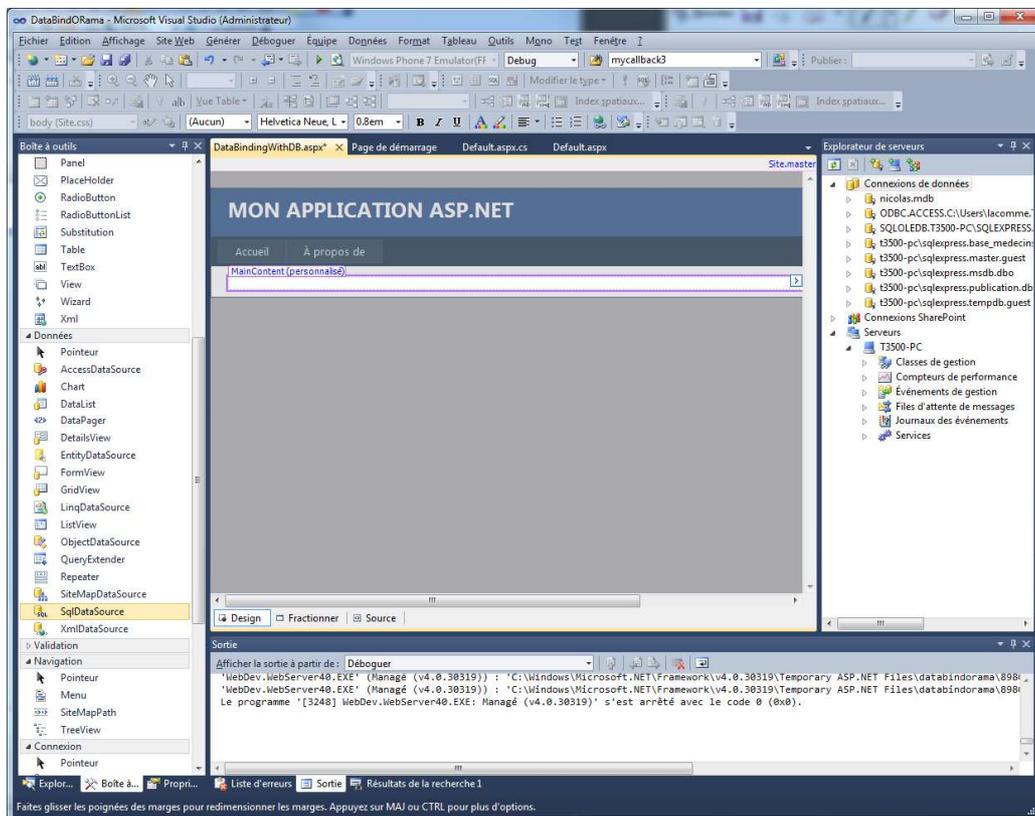


Cette connexion apparait maintenant dans la liste :

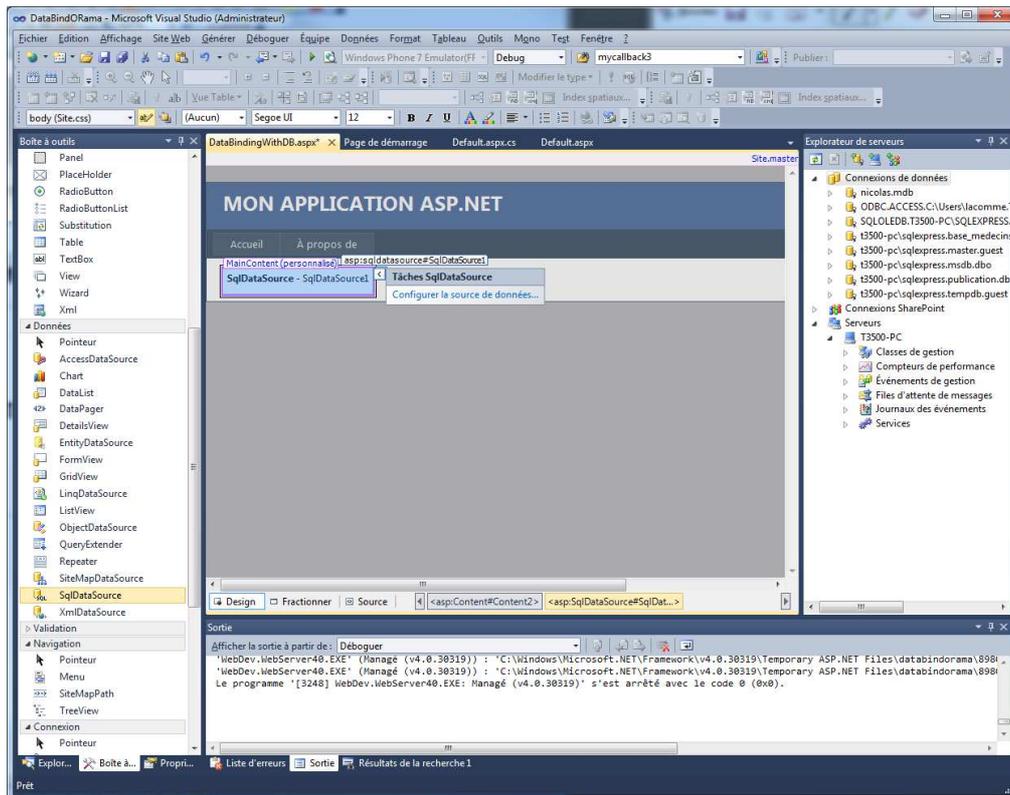


3.4. Manipulation de la base de données

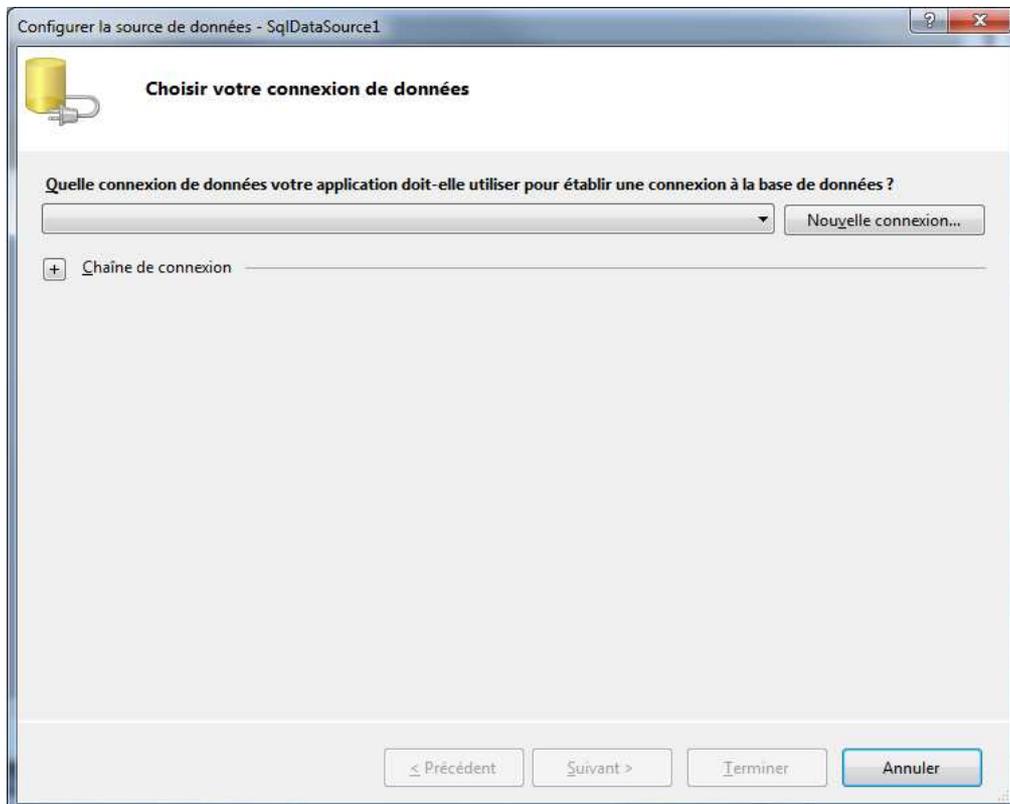
Visualiser la page **DataBindingWithDB.aspx**.



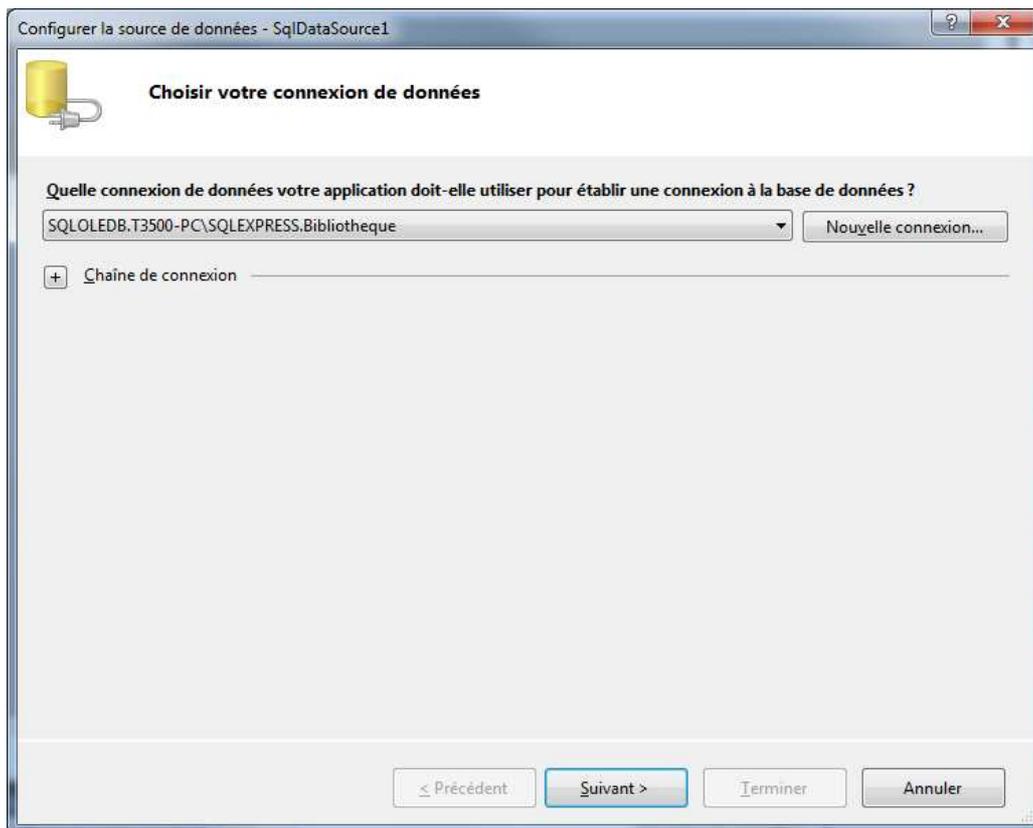
Posez sur la page une **SQLDataSource**.



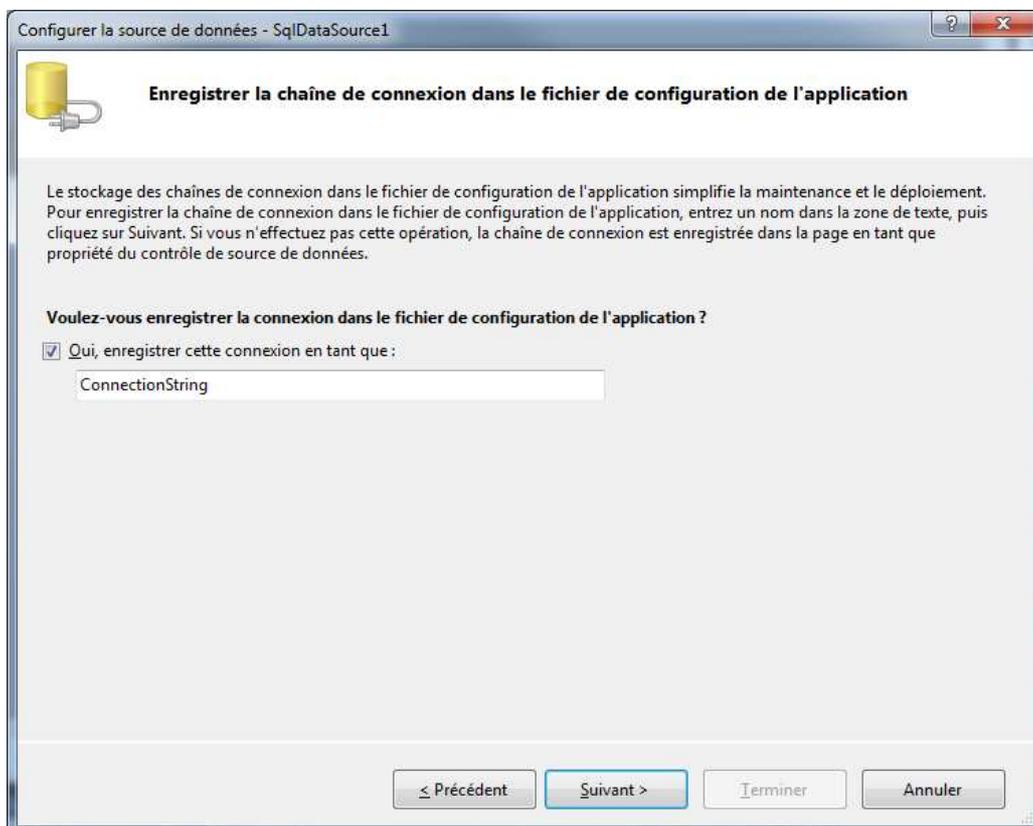
Cliquer droit et choisir ensuite **Configurer la source de données**.



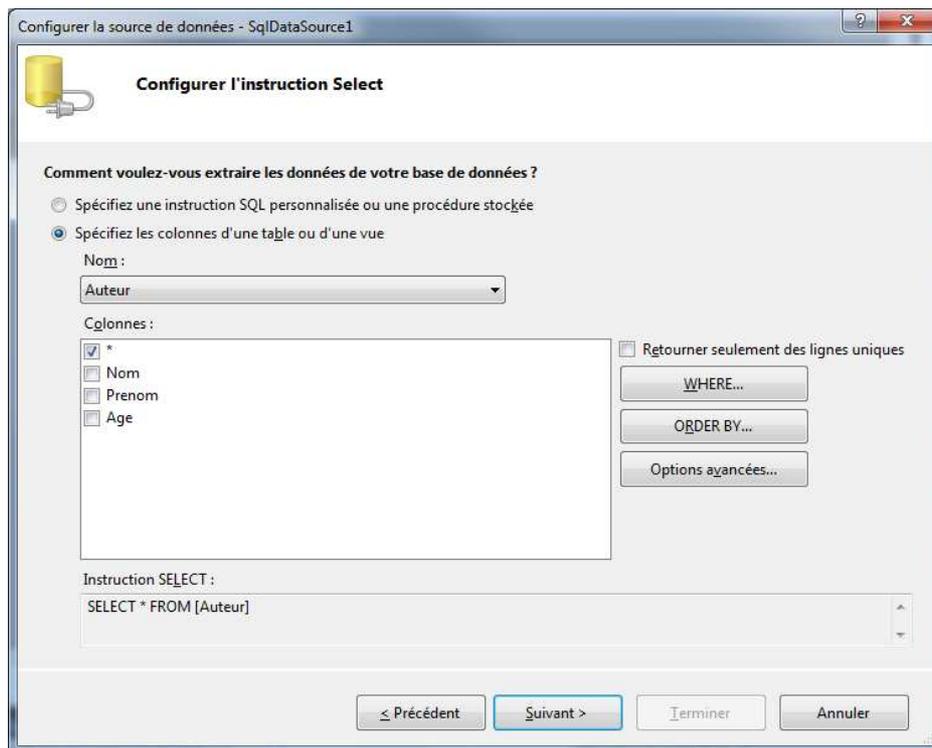
Sélectionnez la **bibliothèque**.



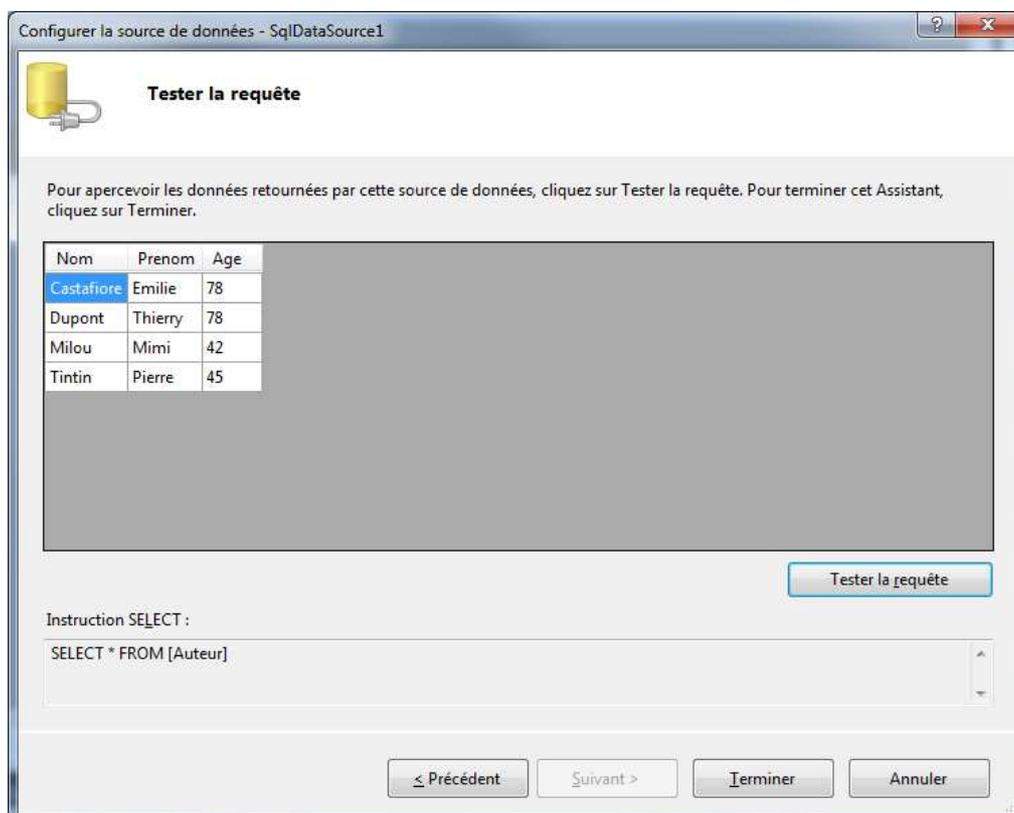
Faire suivant :



Choisir ensuite la table **Auteur**.

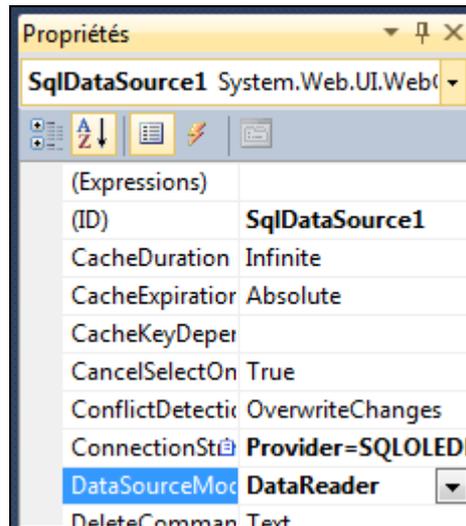


Ce qui donnera :

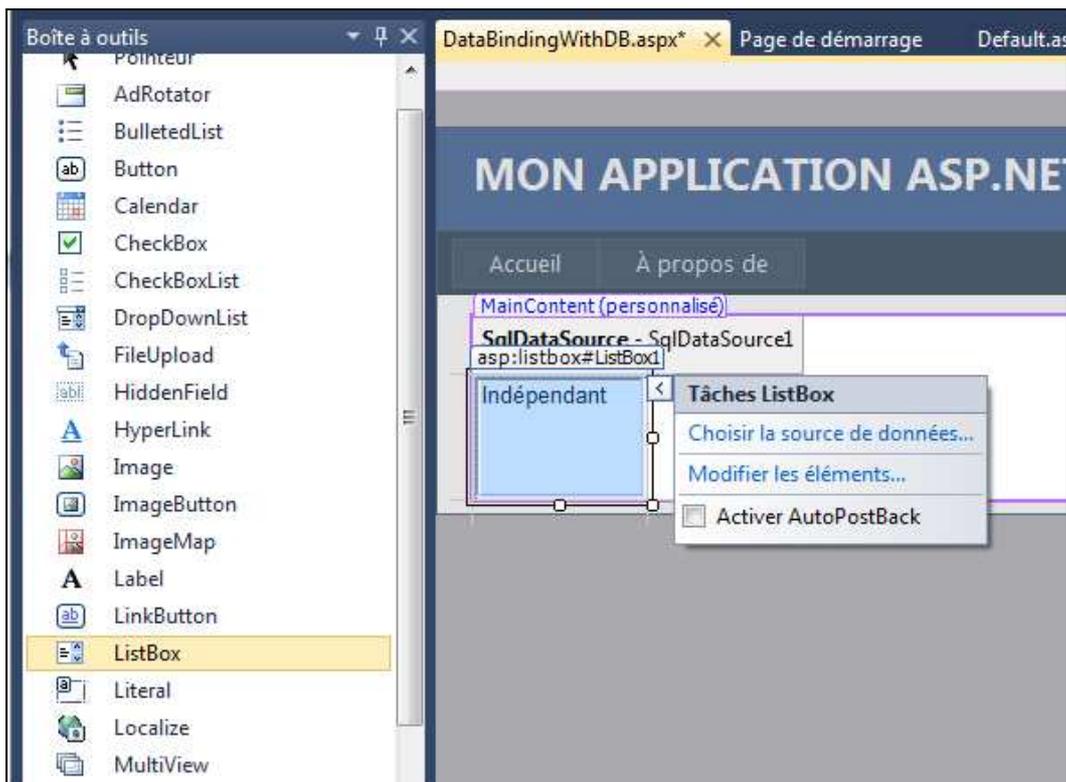


Cliquez ensuite sur le bouton **Terminer**.
Revenez à la page de Visual Studio.

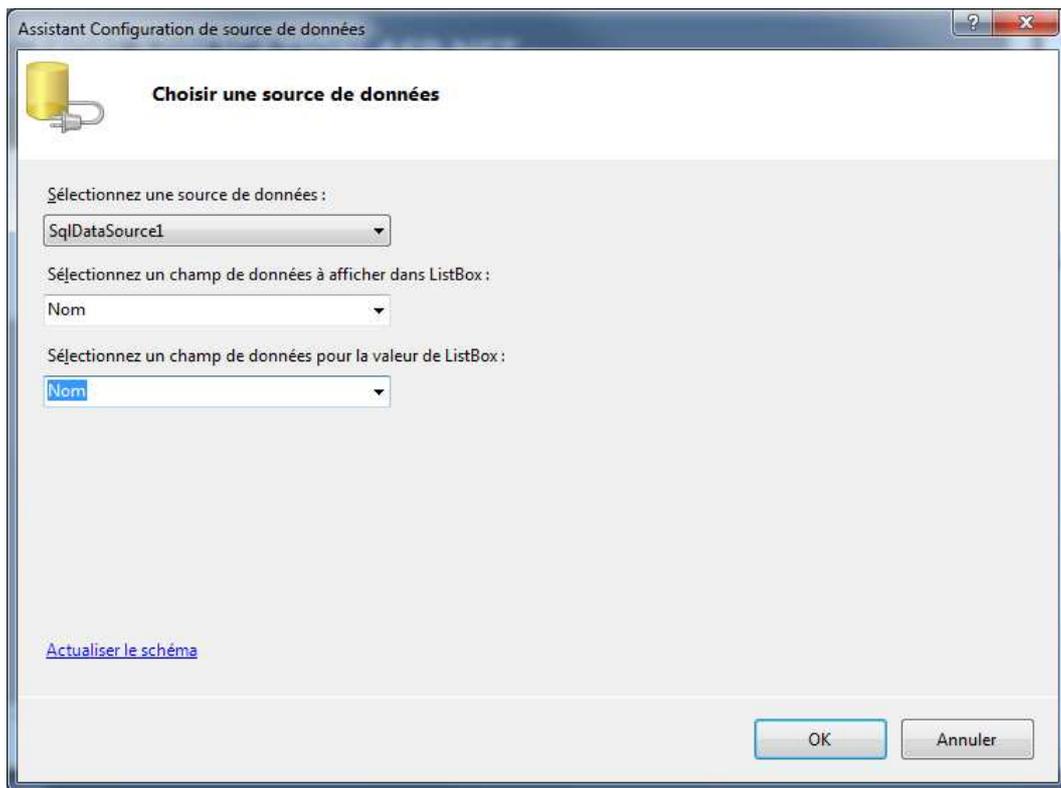
Examinez les attributs de **SqlDataSource1** et modifiez le paramètre **DataSourceMode**.
Passez ce paramètre à la valeur **DataReader**.



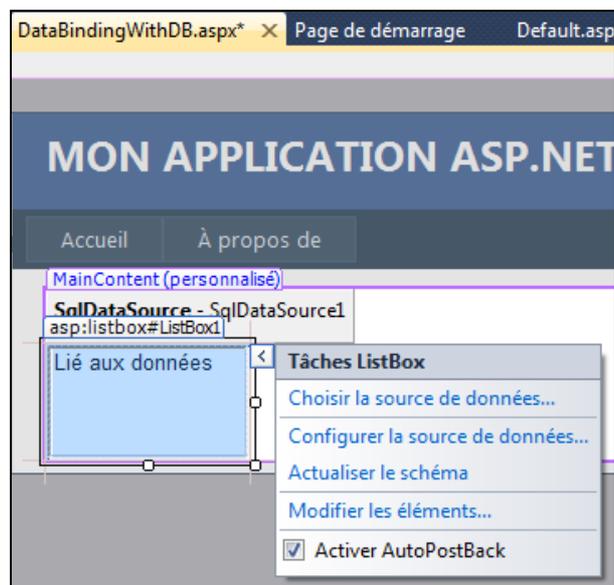
Posez sur la page un contrôle **ListBox**.



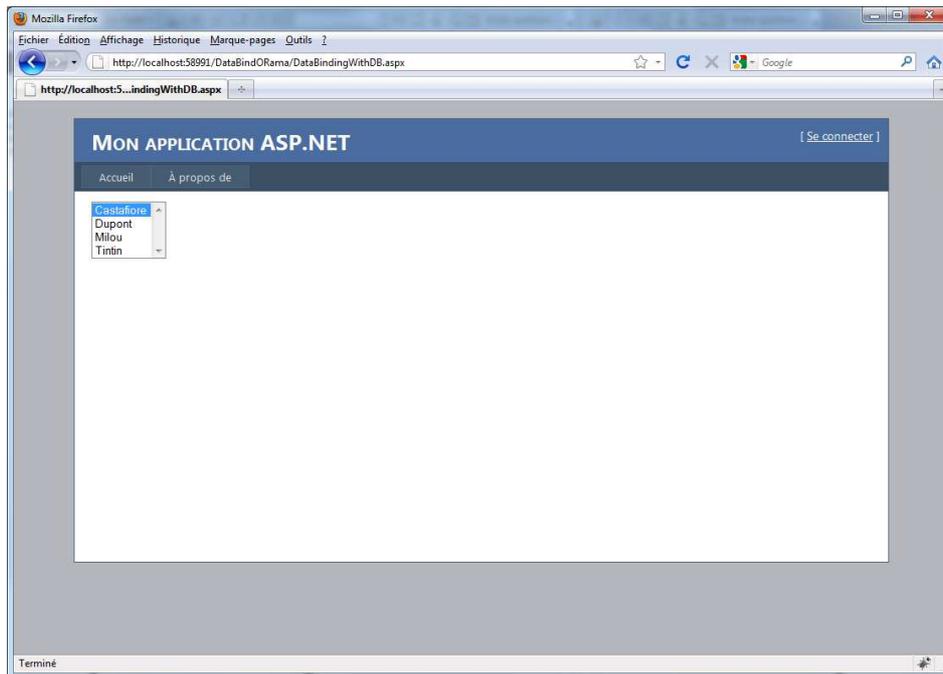
Choisir ensuite la source de données.



Cochez la case **ActiverAutoPostBack**.



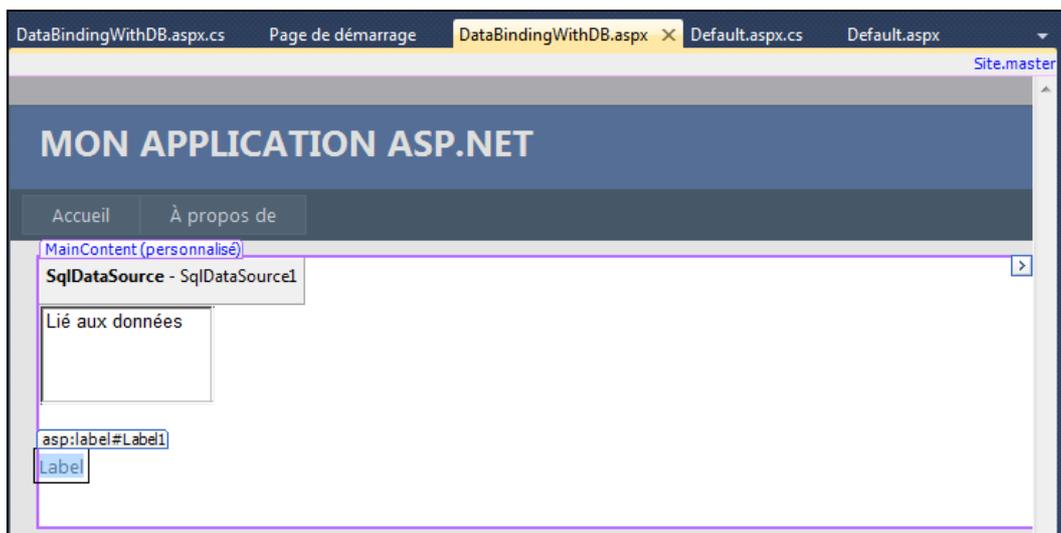
Lancer l'exécution de l'application.



3.5 Manipulation de la base de données : solution « manuelle »

Nous venons d'utiliser les outils automatisés qui génèrent à notre place le code nécessaire. Mais revenons à un mode de programmation plus traditionnel.

Ajouter sous ListBox1 un élément de type label.



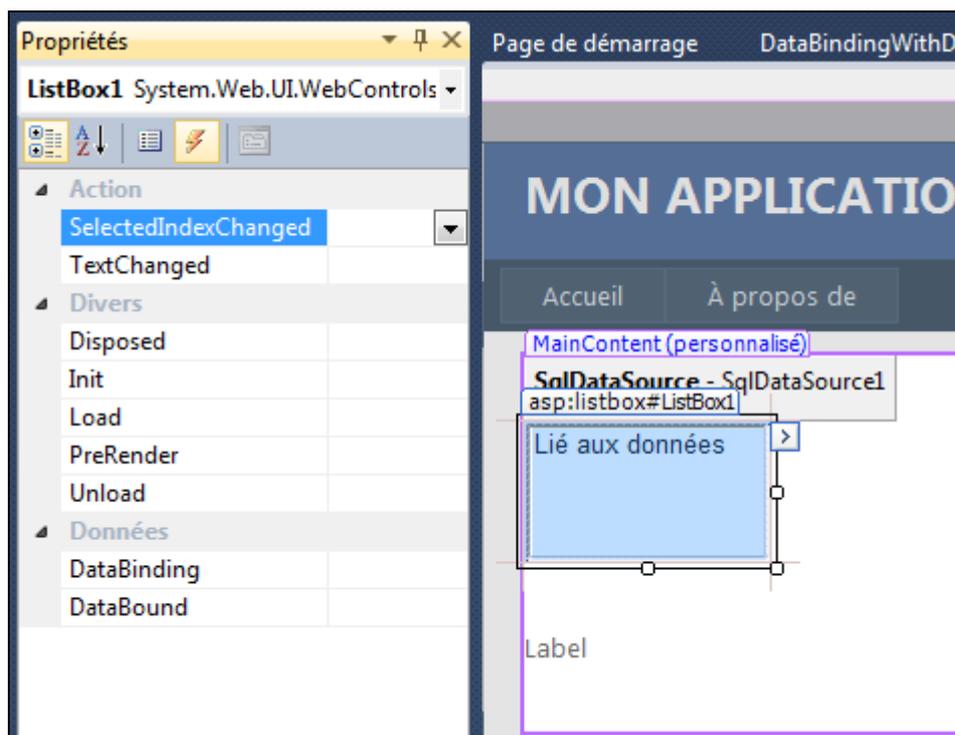
Ouvrir le fichier **DataBindingWithDB.cs**.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class DataBindingWithDB : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
}
```

Le code ne contient pour l'instant aucun élément permettant de gérer les actions de l'utilisateur.

Sélectionnez Listbox1 à la souris et examiner ses propriétés ainsi que les évènements. Choisir **SelectedIndexChanged**.



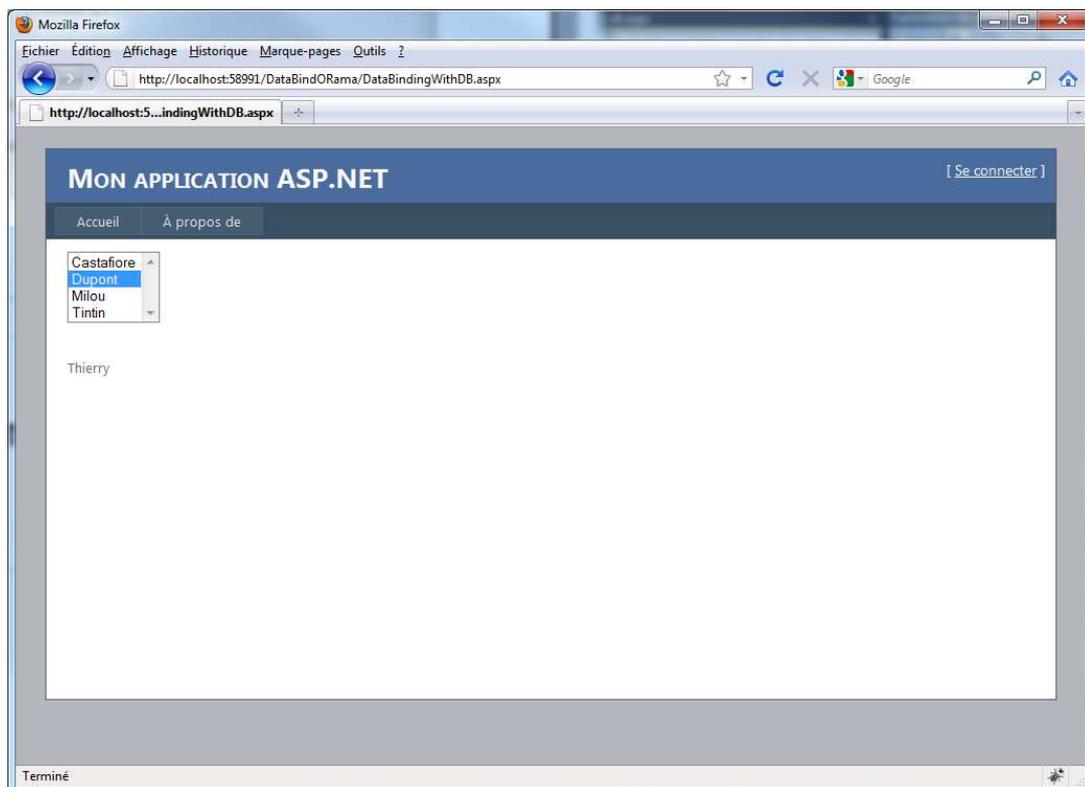
Faire un double click sur la zone à droite de l'action **SelectedIndexChanged**.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class DataBindingWithDB : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (this.IsPostBack == false)
        {
            this.ListBox1.DataTextField = "Nom";
            this.ListBox1.DataValueField = "Prenom";
            this.ListBox1.DataBind();
        }
    }

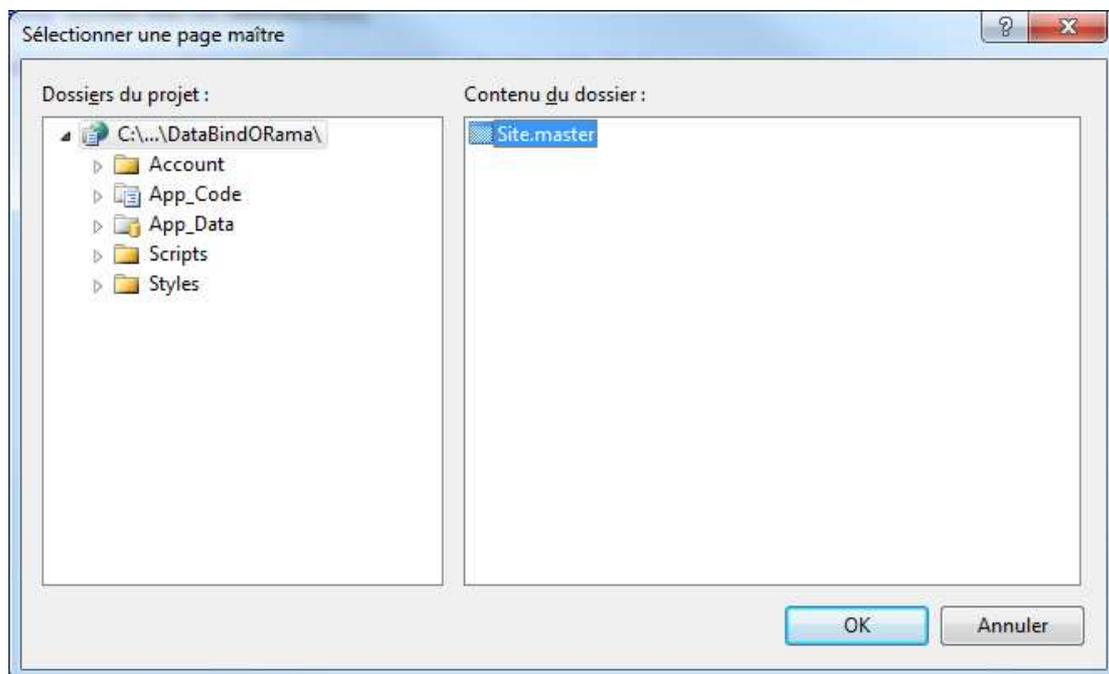
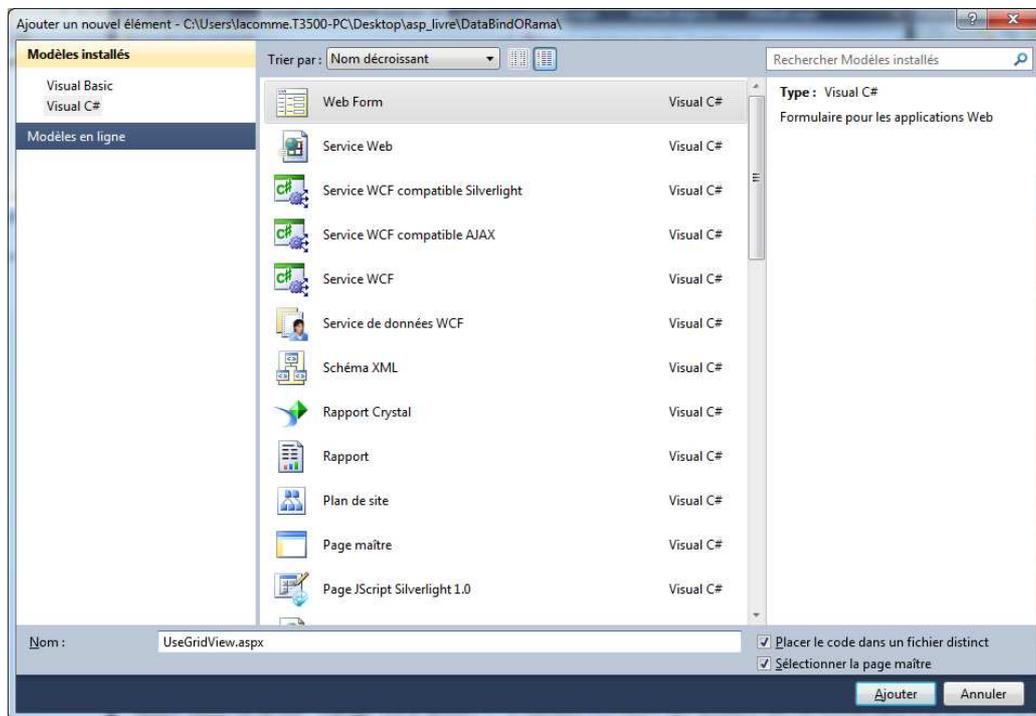
    protected void ListBox1_SelectedIndexChanged(object sender, EventArgs e)
    {
        this.Label1.Text = this.ListBox1.SelectedValue;
    }
}
```

Cela donne à l'exécution :

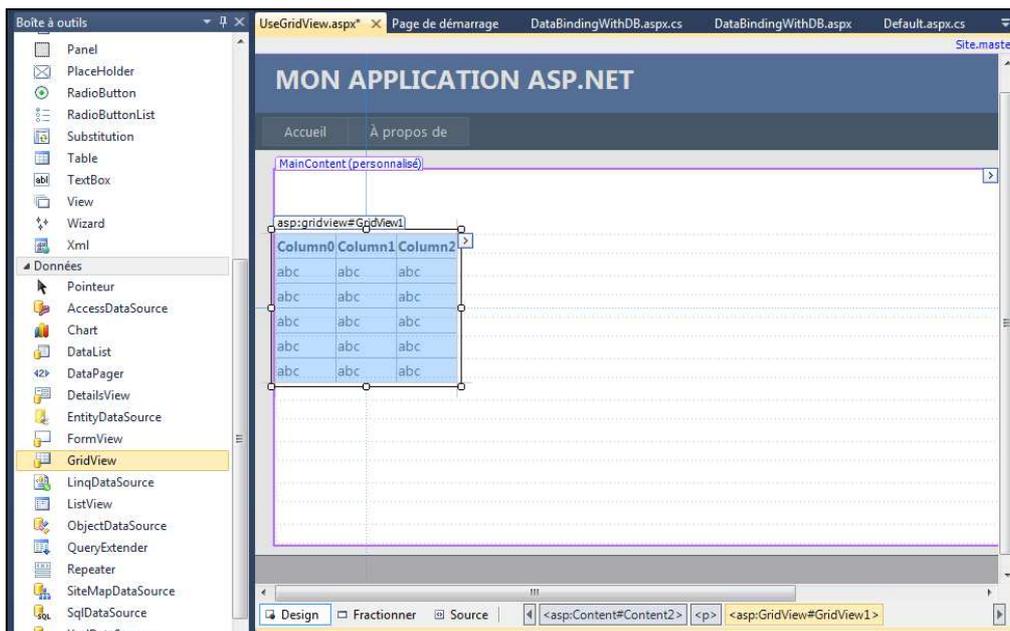


3.6. Utilisation des contrôles « GridView ».

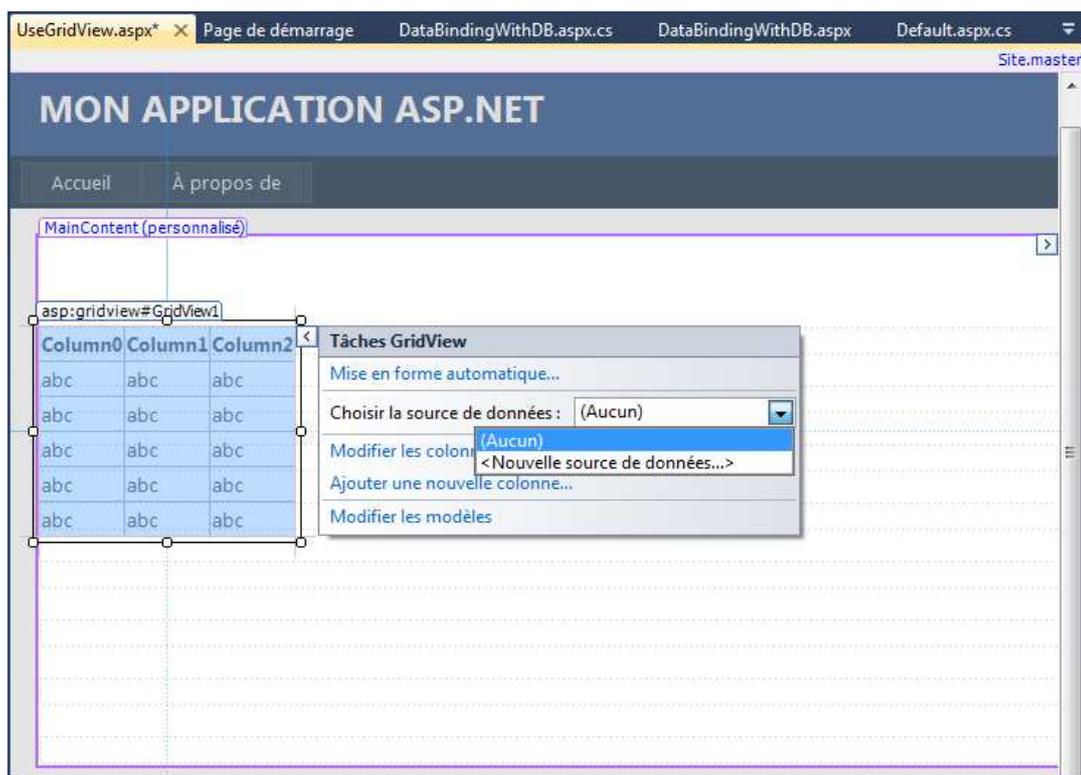
Ajouter un nouveau formulaire nommé **UseGridView**.



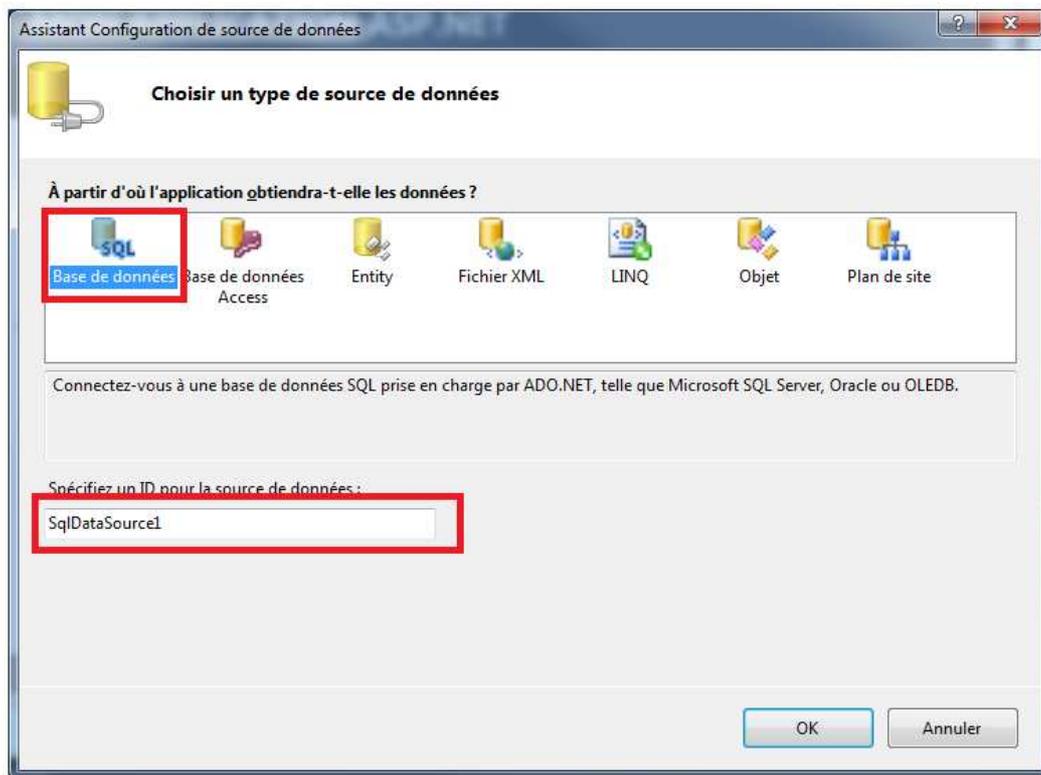
Posez un élément GridView sur la page **UseGridView.aspx**.



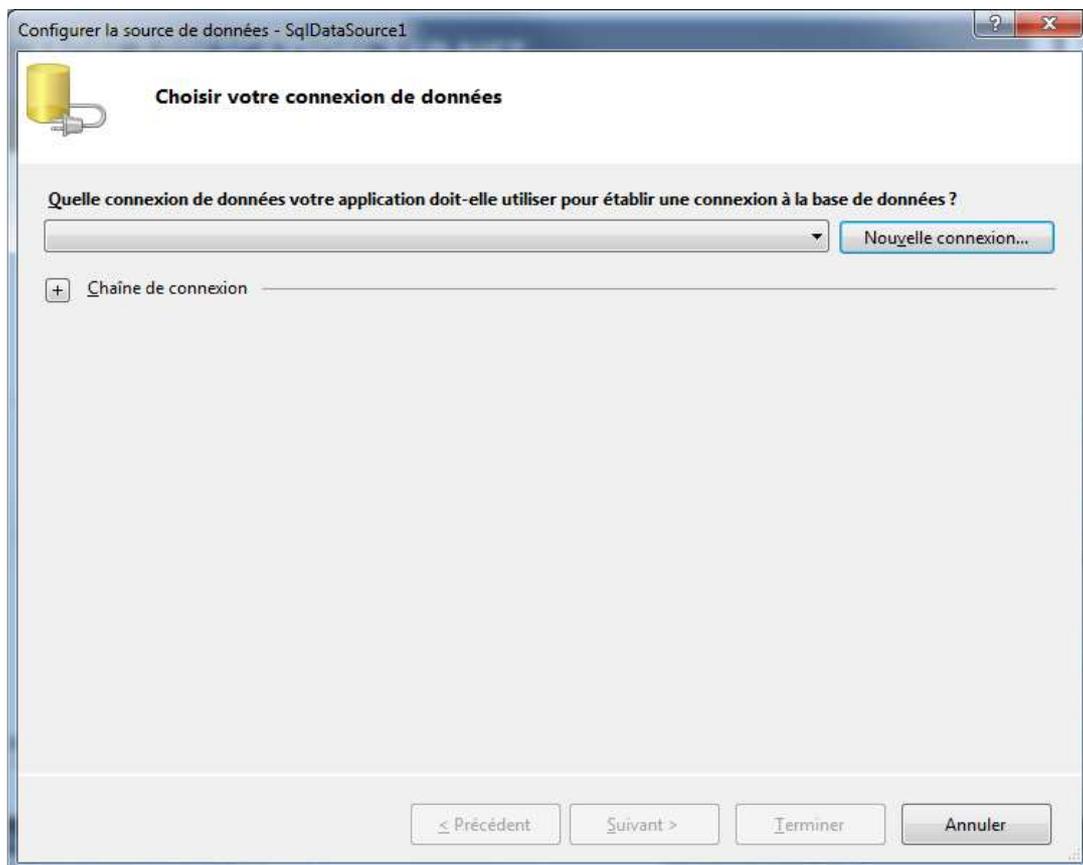
Allez ensuite sur **Nouvelle source de données**.



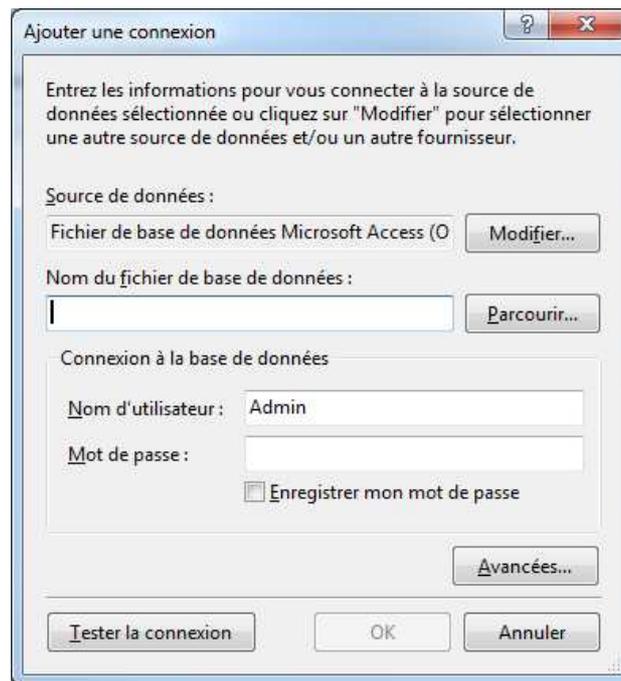
Choisir **Base de données SQL**.



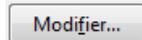
Cela ouvre automatique l'assistant de connexion.



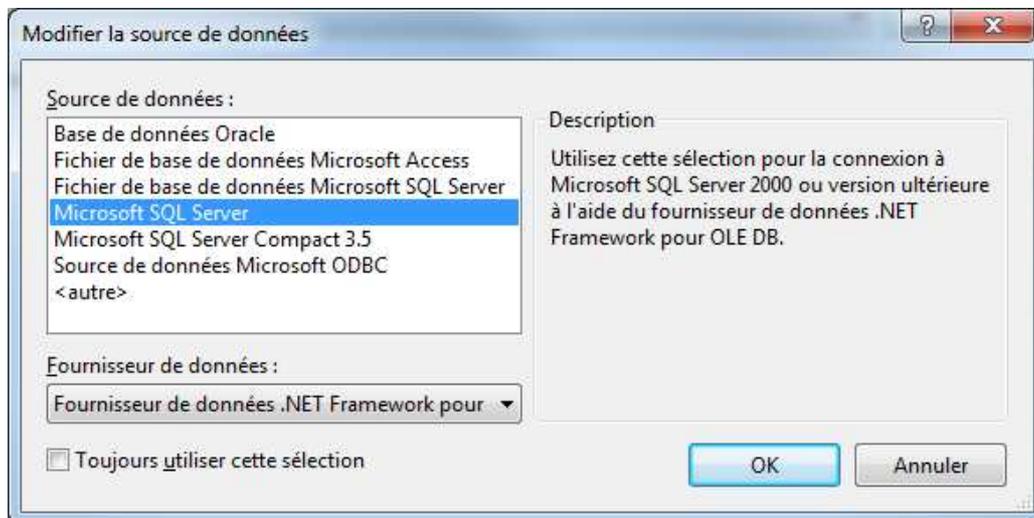
Faire Nouvelle Connexion.



Choisir le bouton



Choisir **Microsoft SQL Server**.



Ajouter une connexion

Entrez les informations pour vous connecter à la source de données sélectionnée ou cliquez sur "Modifier" pour sélectionner une autre source de données et/ou un autre fournisseur.

Source de données :
Microsoft SQL Server (OLE DB) Modifier...

Nom du serveur :
T3500-PC\SQLEXPRESS Actualiser

Connexion au serveur

Utiliser l'authentification Windows
 Utiliser l'authentification SQL Server

Nom d'utilisateur :
Mot de passe :
 Enregistrer mon mot de passe

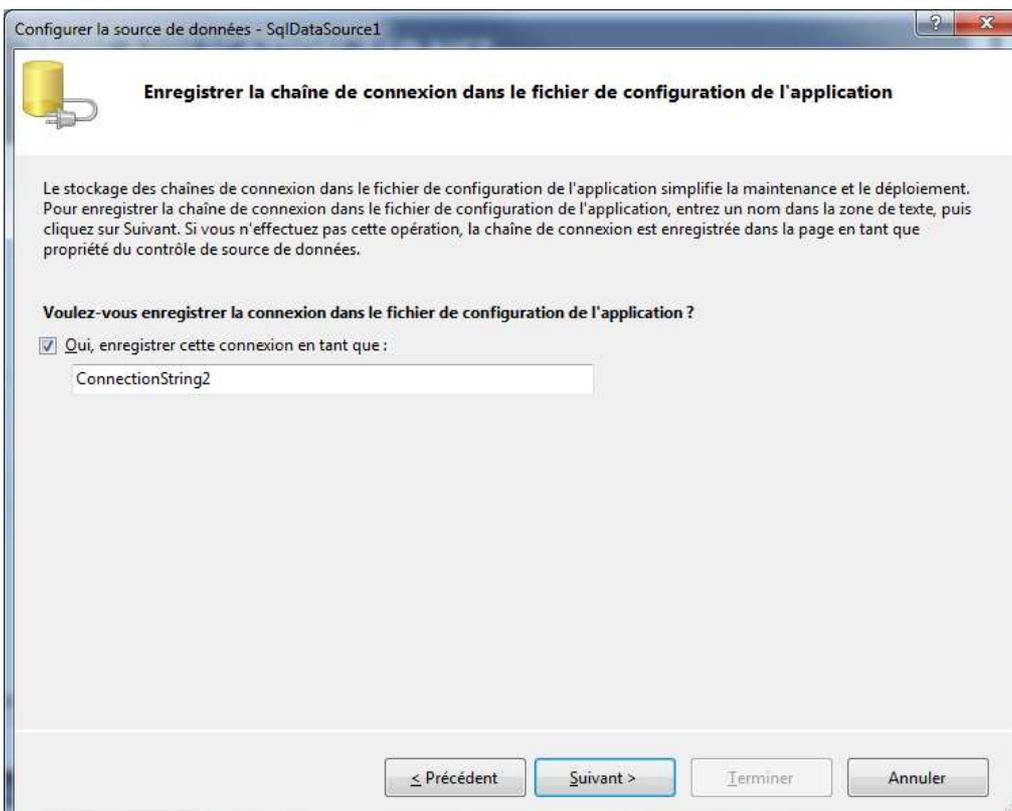
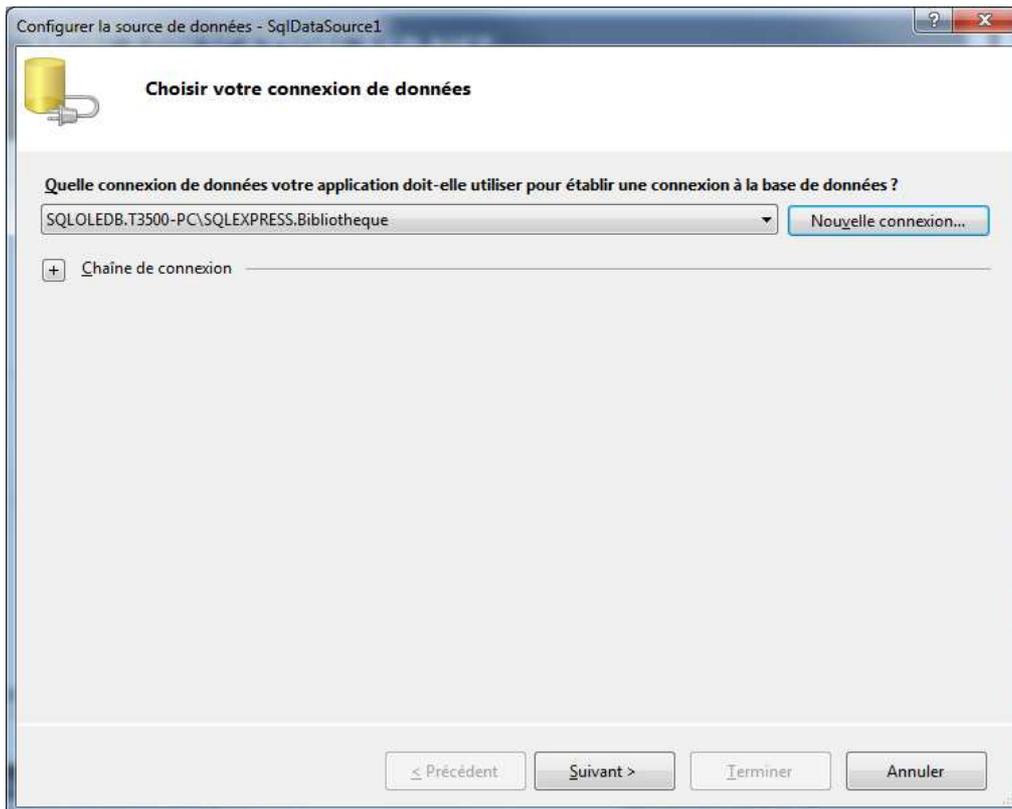
Connexion à la base de données

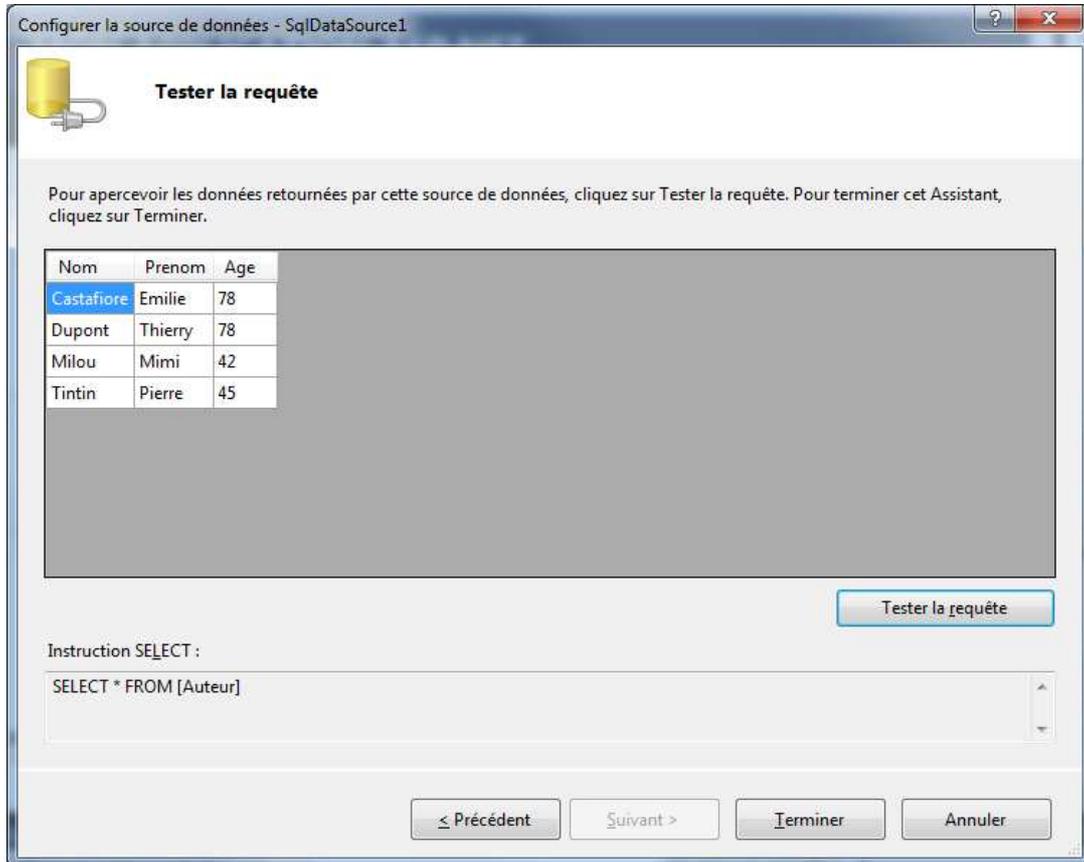
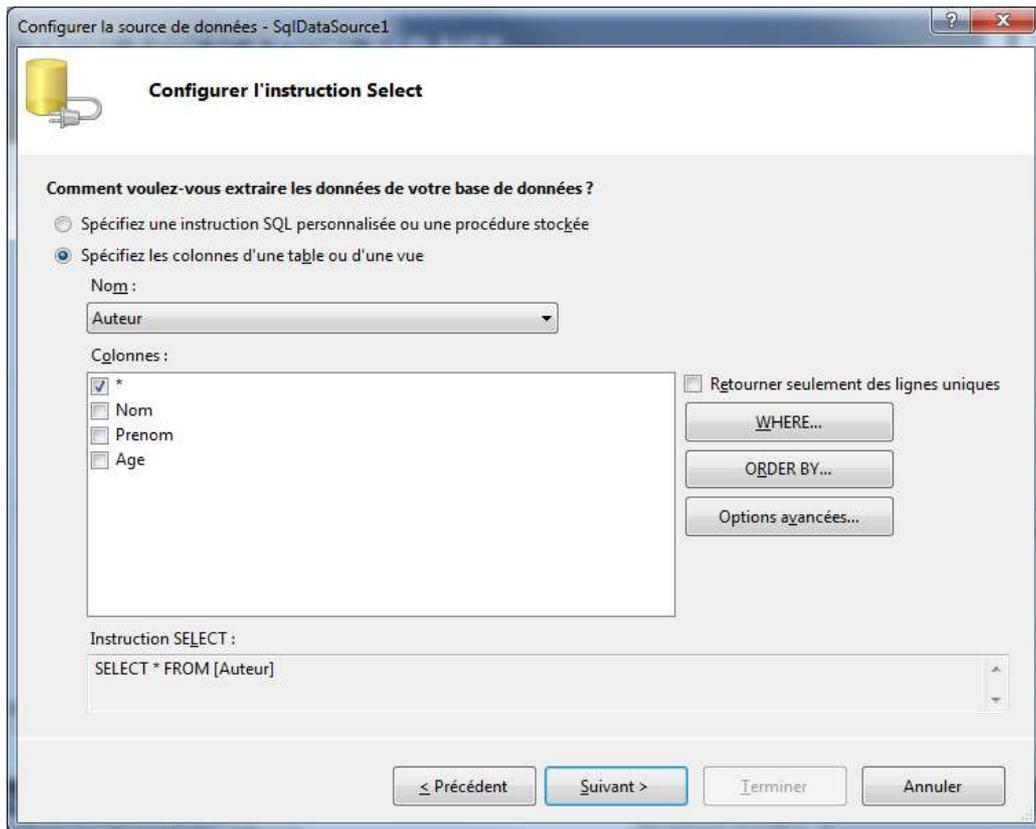
Sélectionner ou entrer un nom de base de données :
Bibliotheque ▼

Attacher un fichier de base de données :
 Parcourir...
Nom logique :

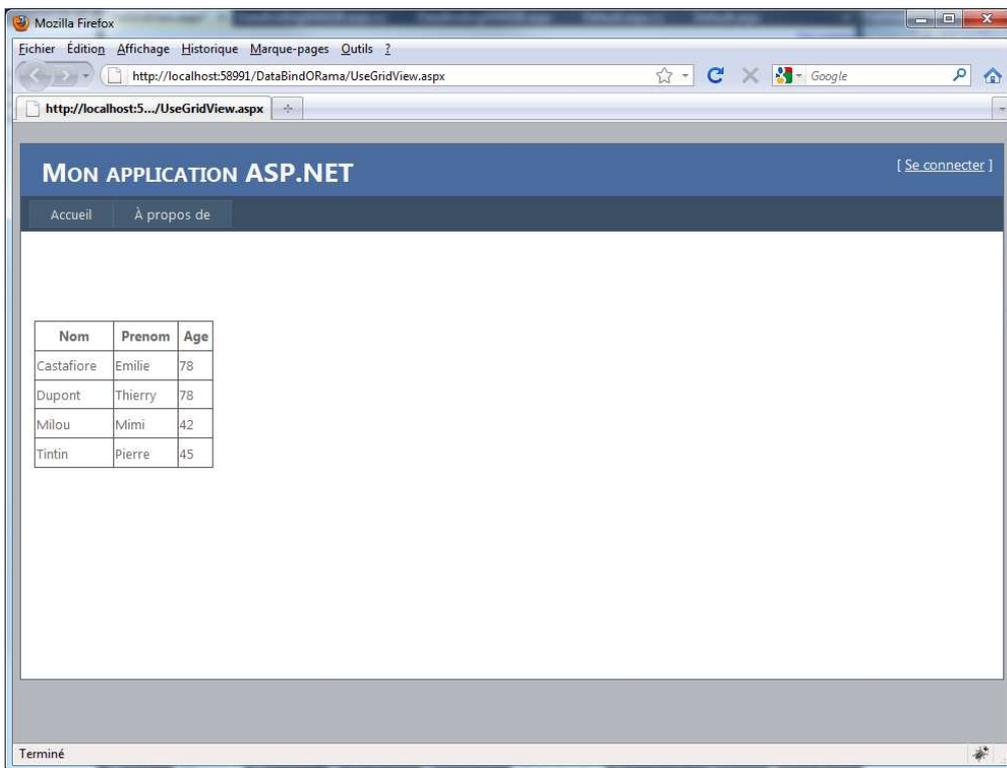
Avancées...

Tester la connexion OK Annuler



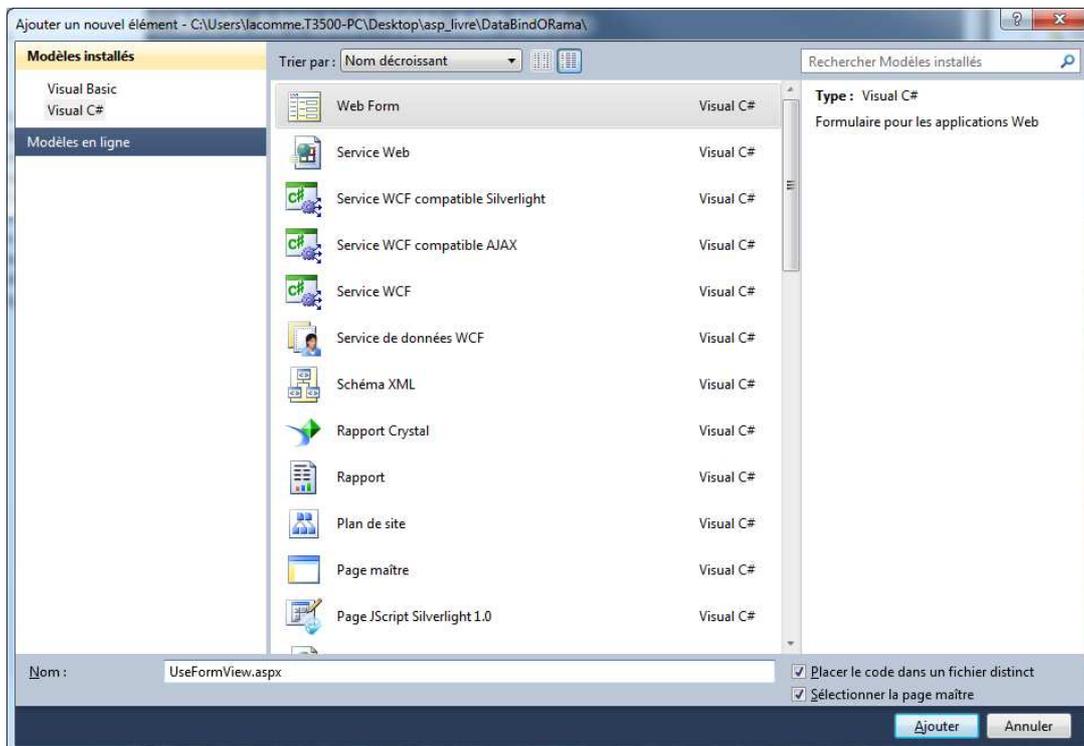


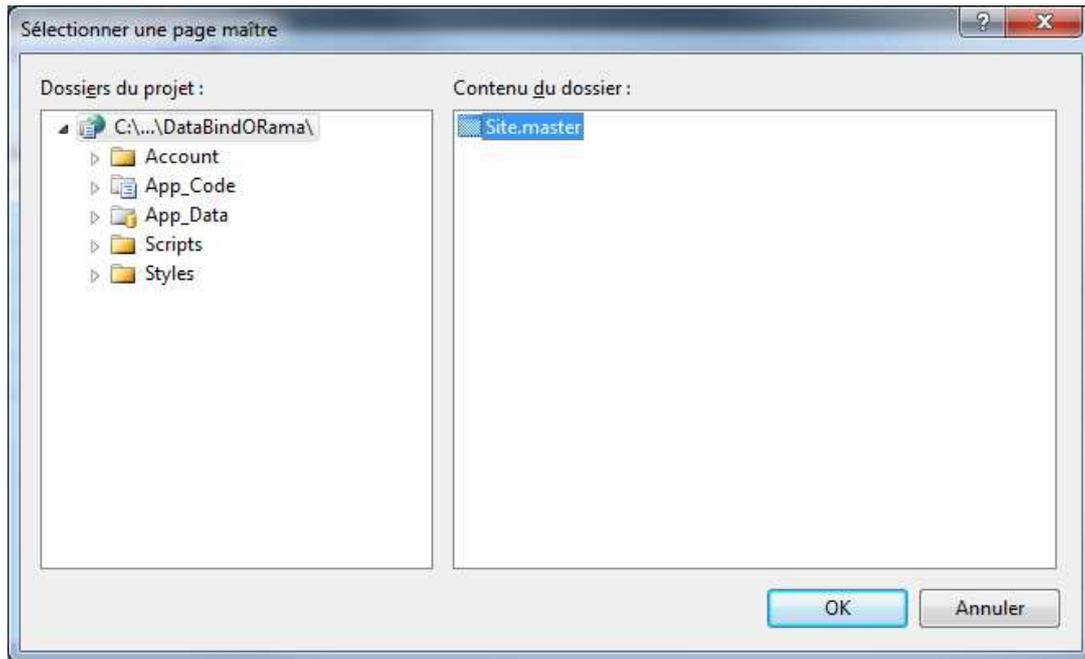
Ceci donne un affichage très flatteur comme celui-ci :



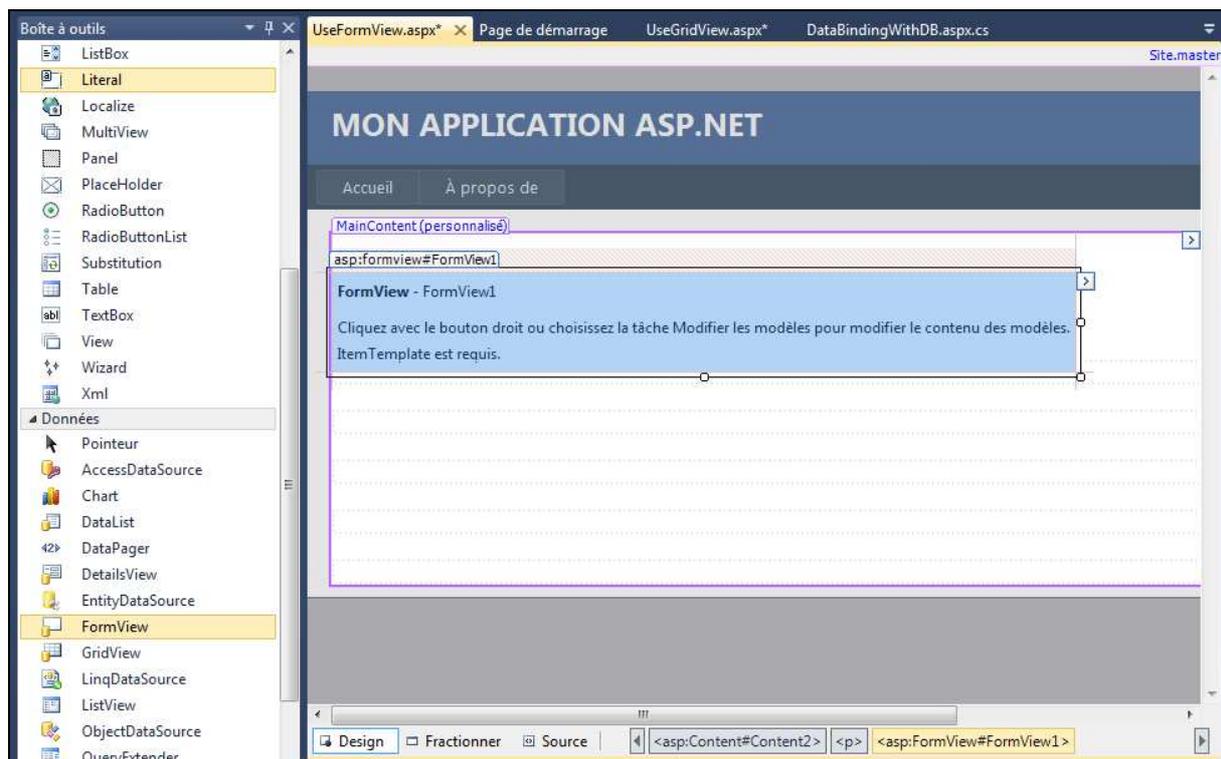
3.7. Utilisation des contrôles « FormView ».

Ajouter un nouveau formulaire nommé UseFormView.

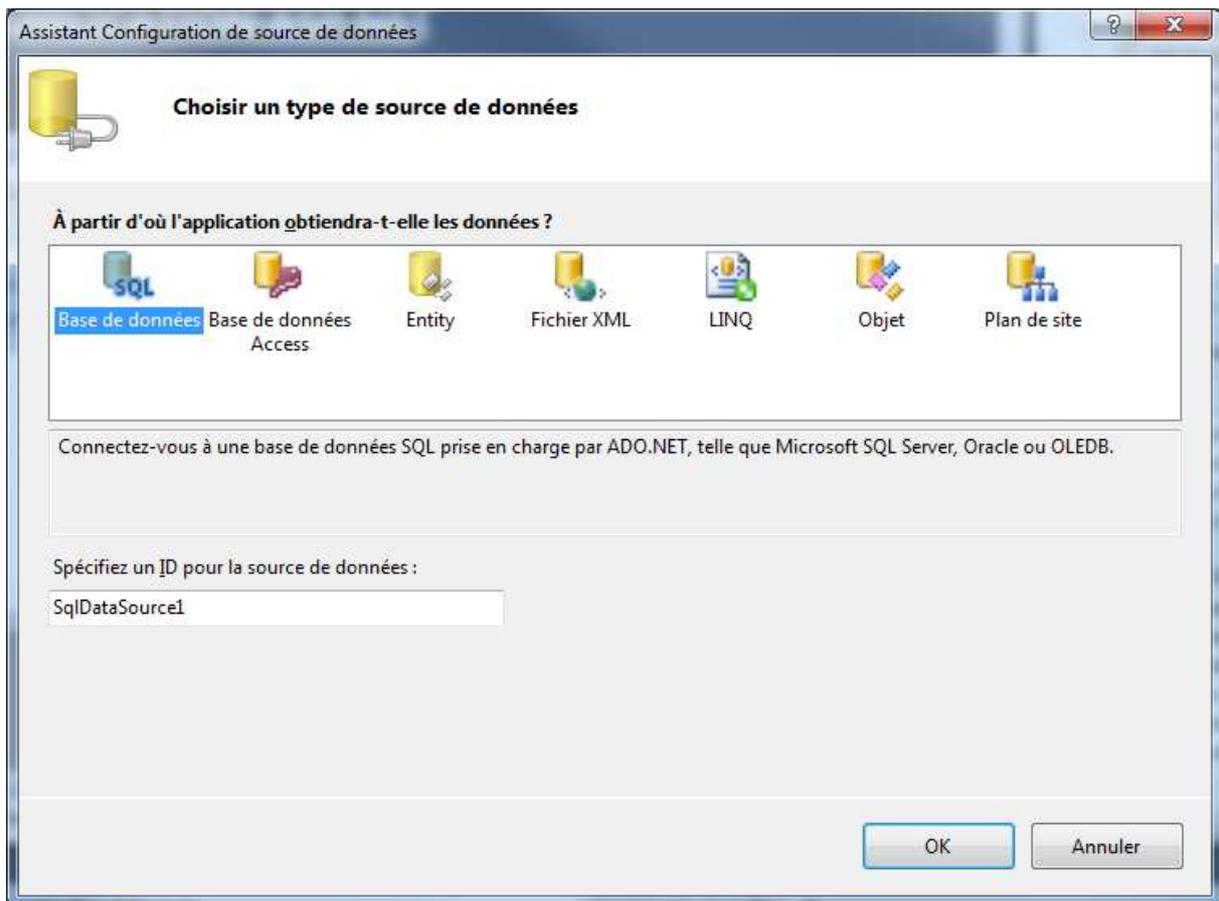
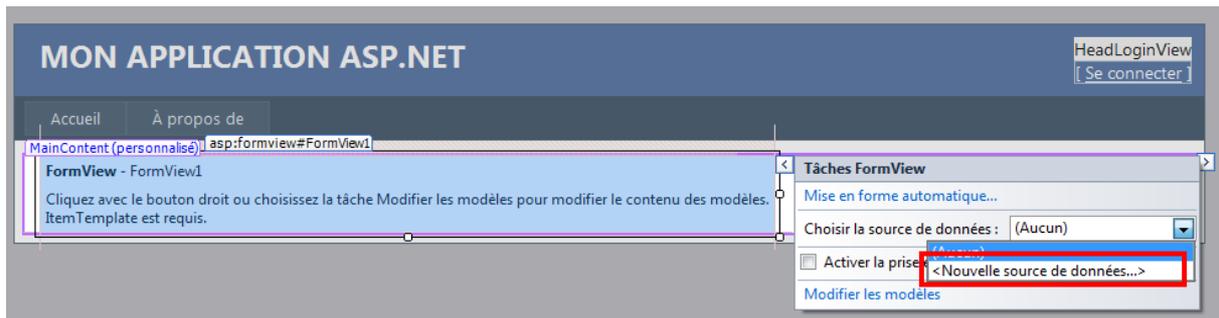


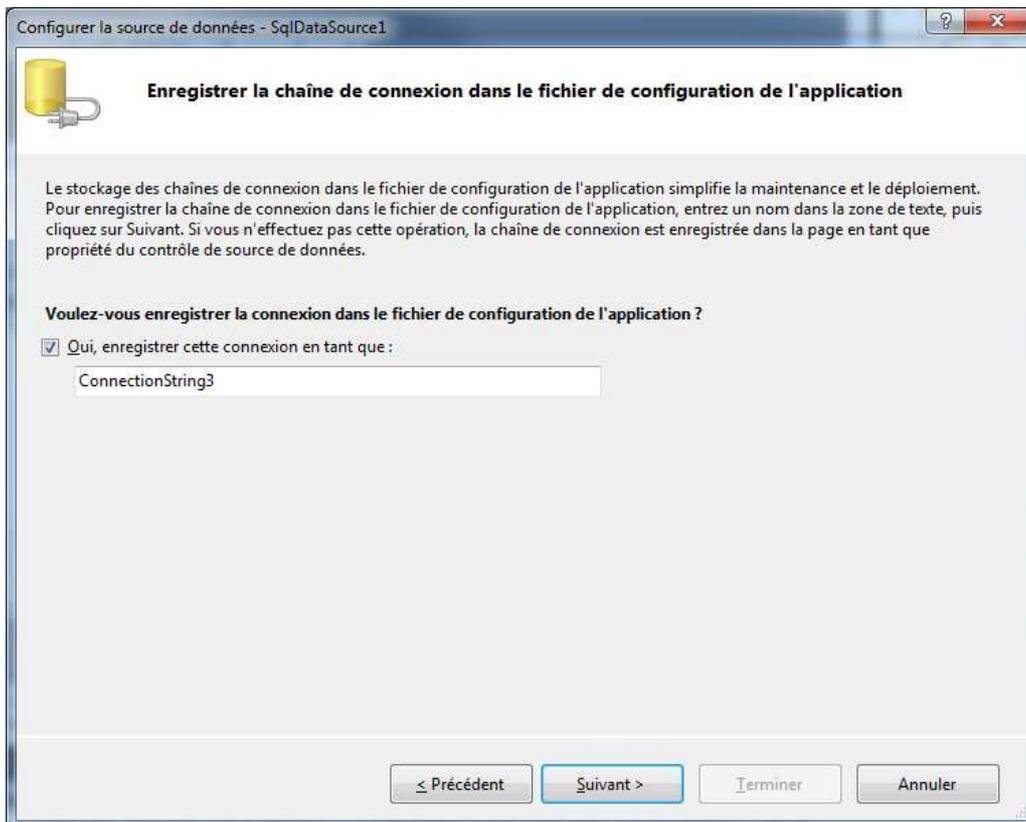
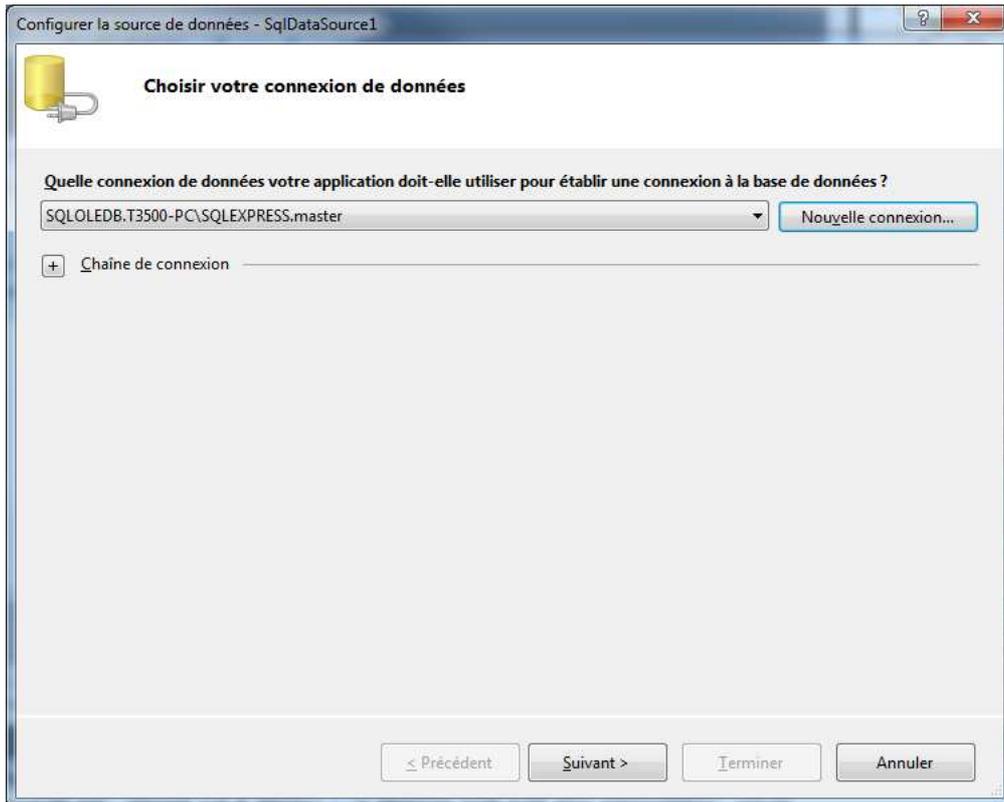


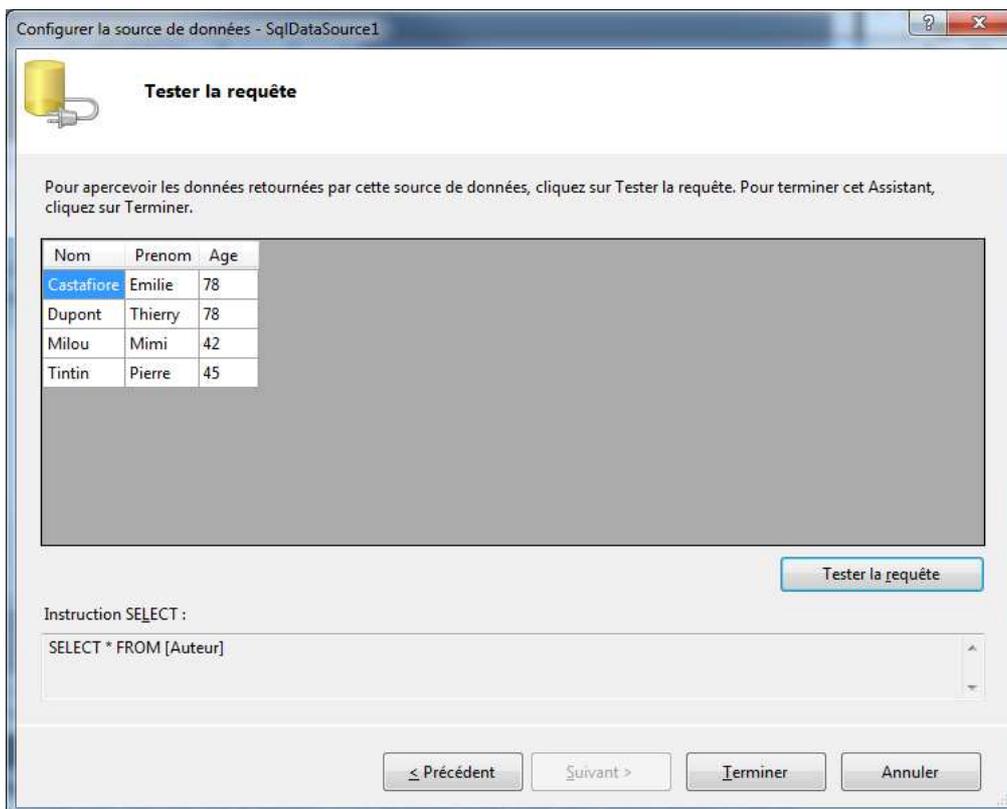
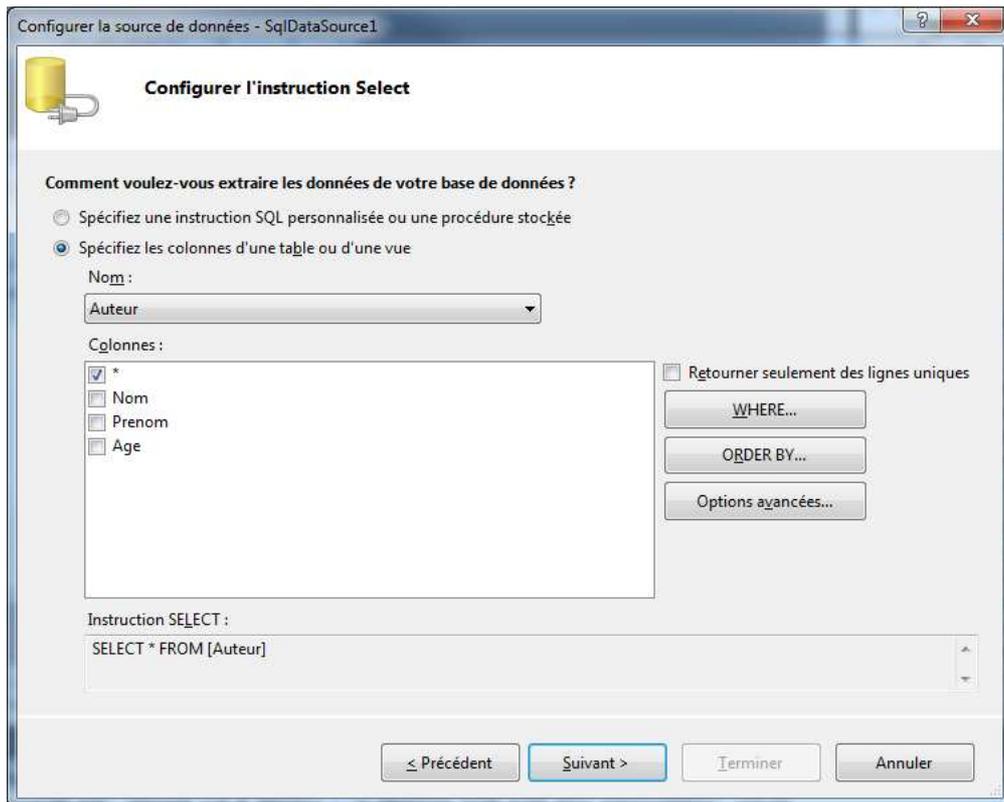
Ajouter sur la page un contrôle **FormView**.



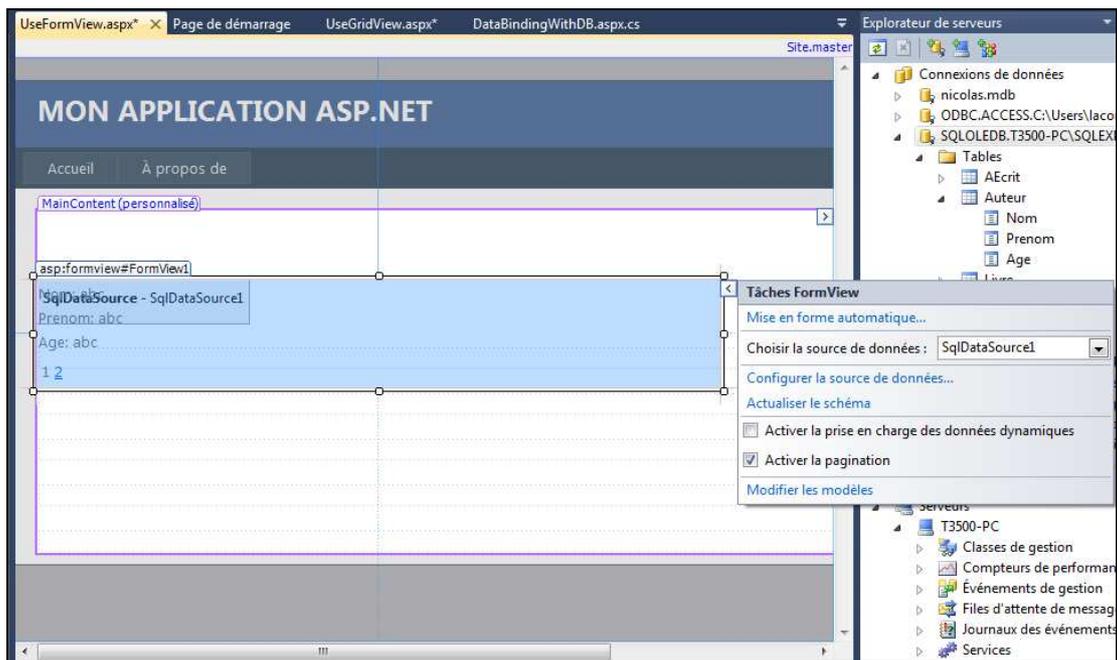
Allez ensuite sur **Nouvelle source de données** et procéder comme dans la section précédente.



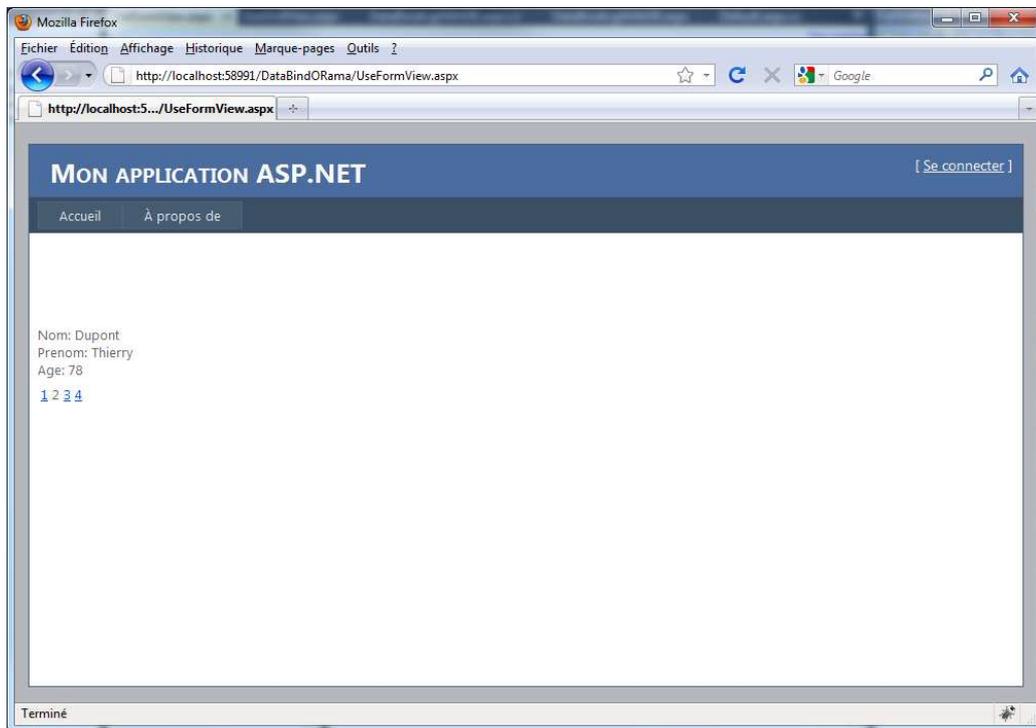




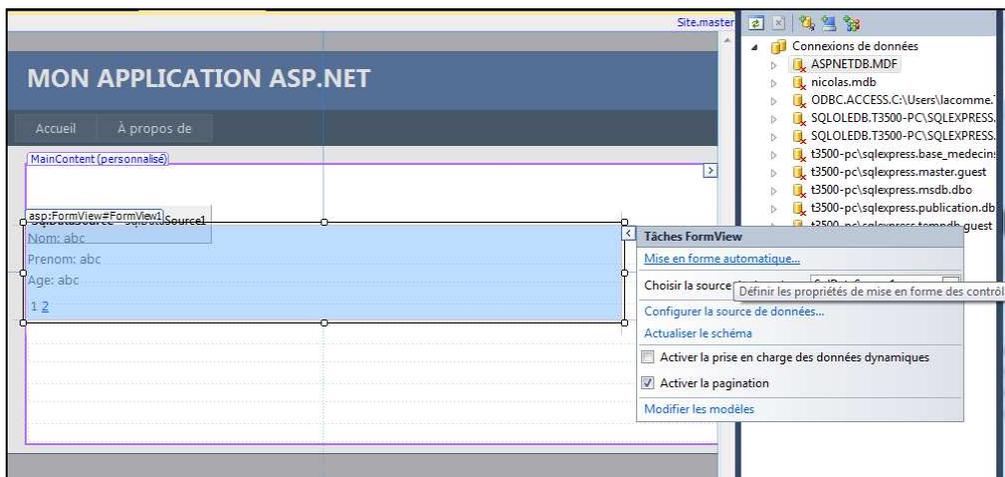
Une fois revenu à la forme, activez la pagination comme ci-dessous :



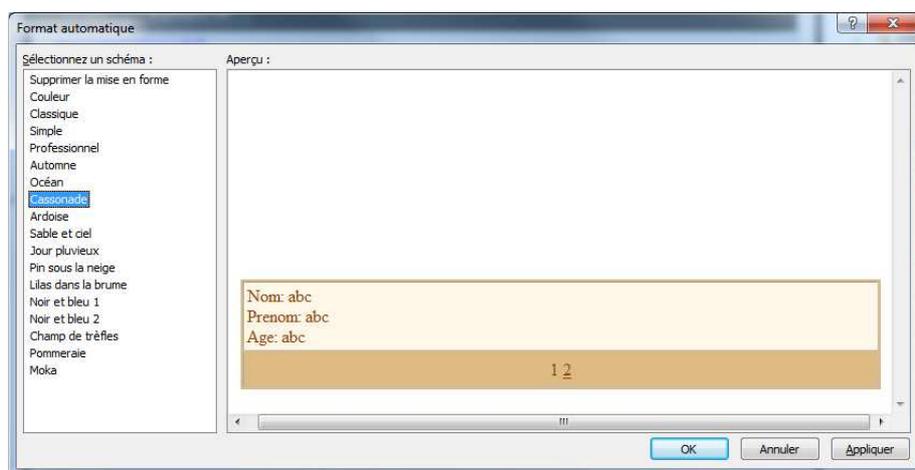
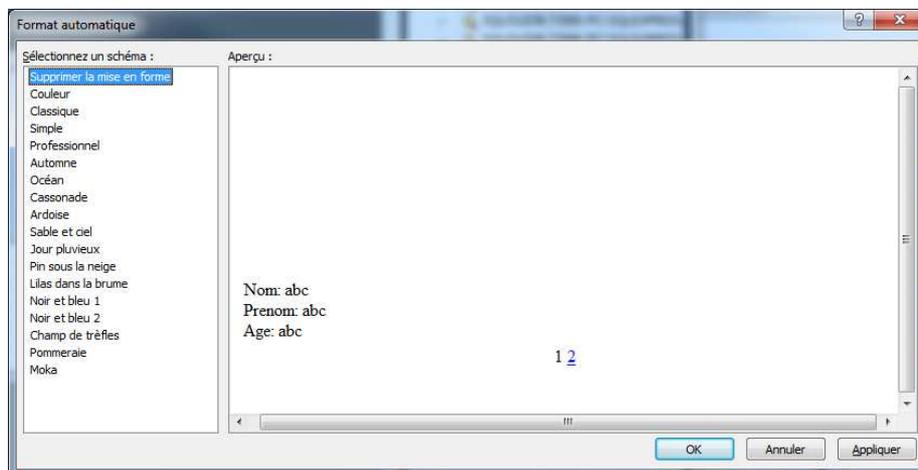
Ceci donne à l'exécution :



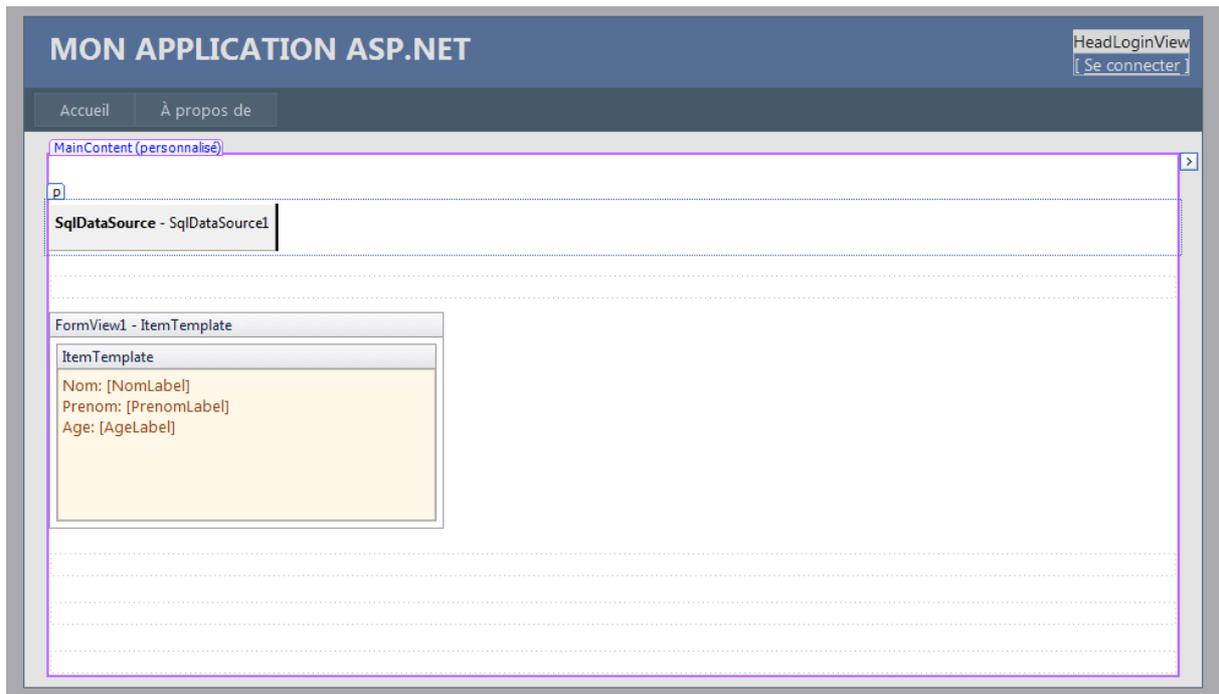
Vous pouvez modifier le modèle en utilisant le lien **Mise en forme automatique**.



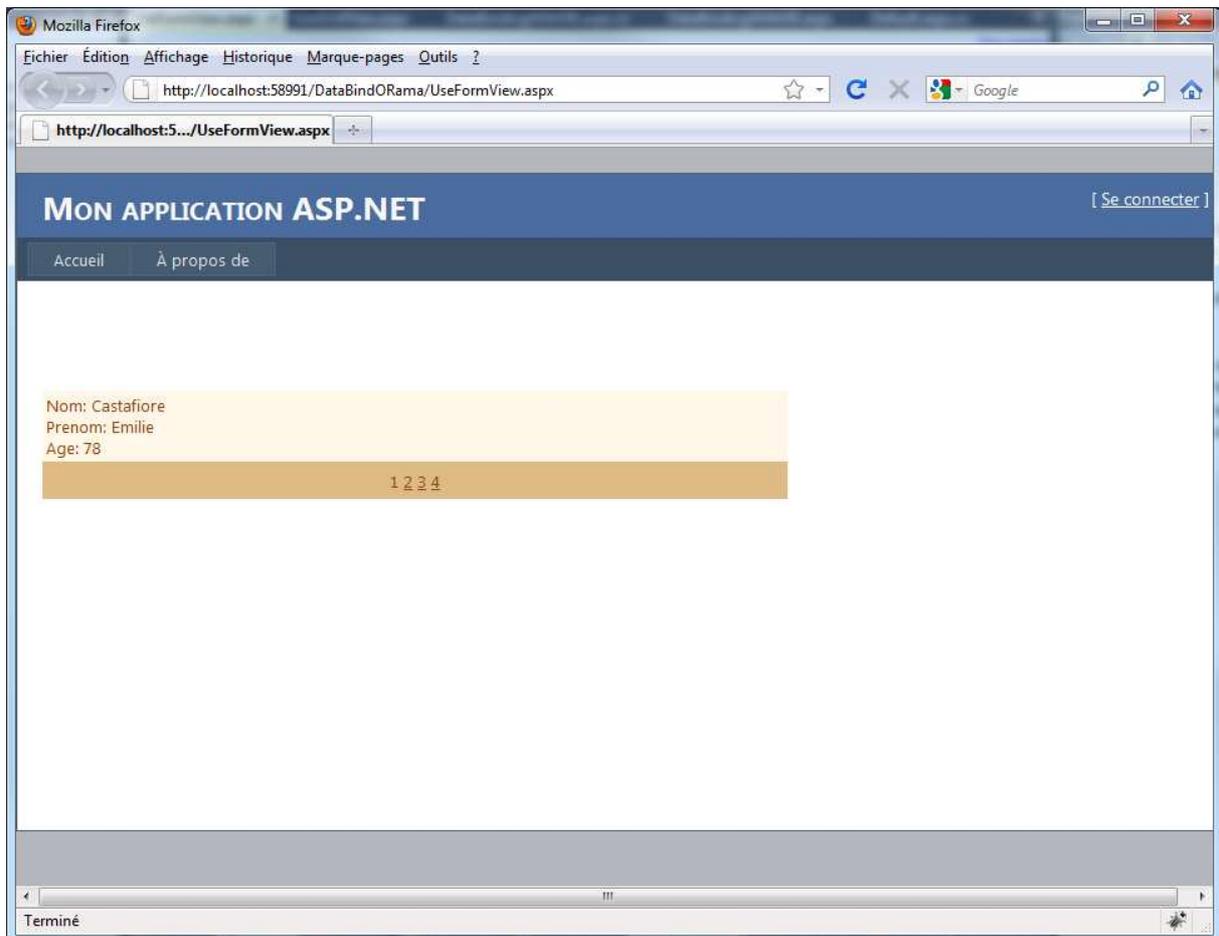
Ce qui permet de choisir la mise en page :



Profitez-en pour répartir correctement les contrôles sur la page comme suit :

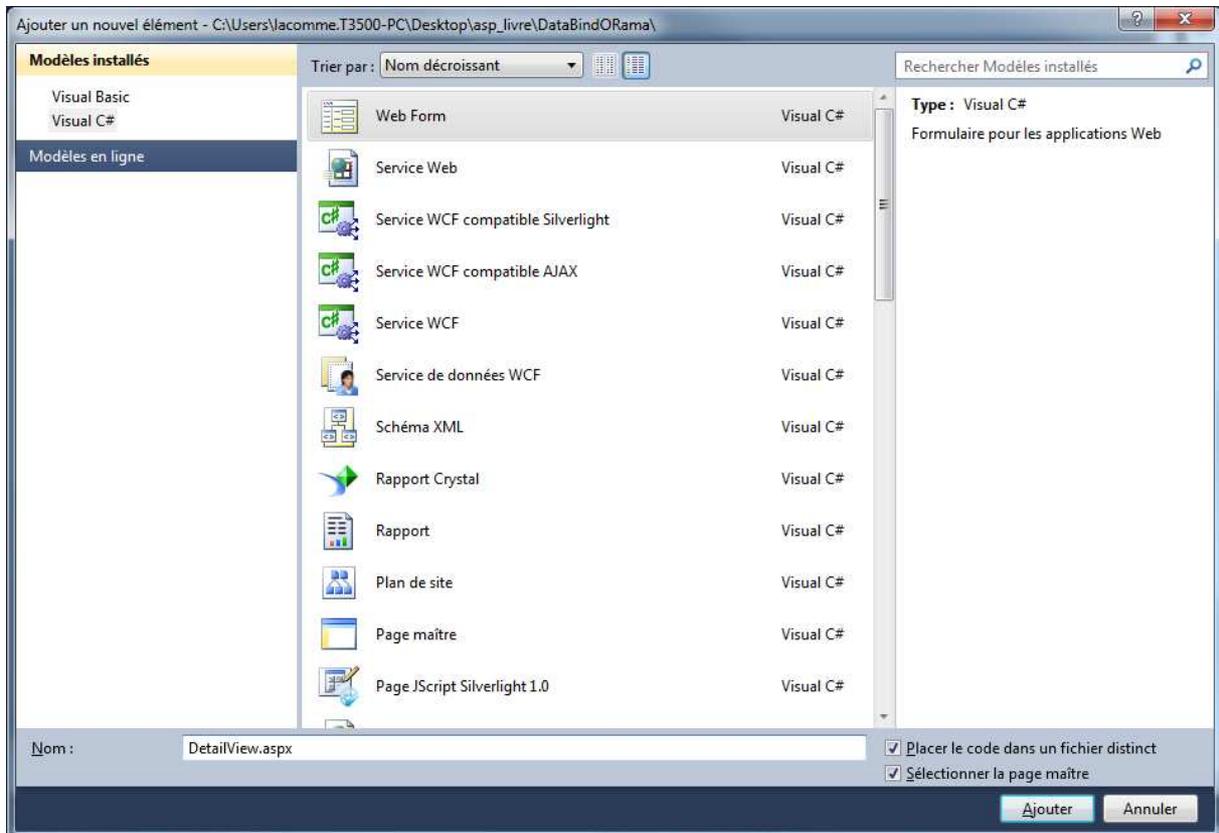


Cela donne un affichage plus flatteur.

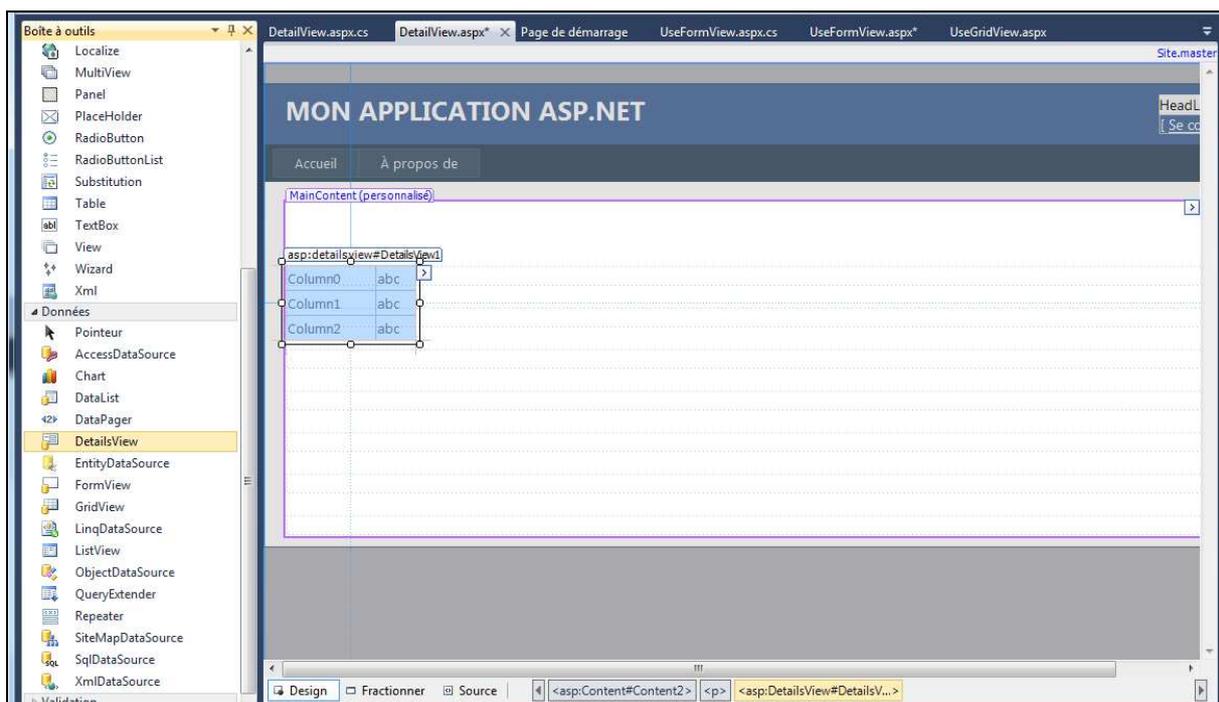


3.8. Utilisation des contrôles « DetailView ».

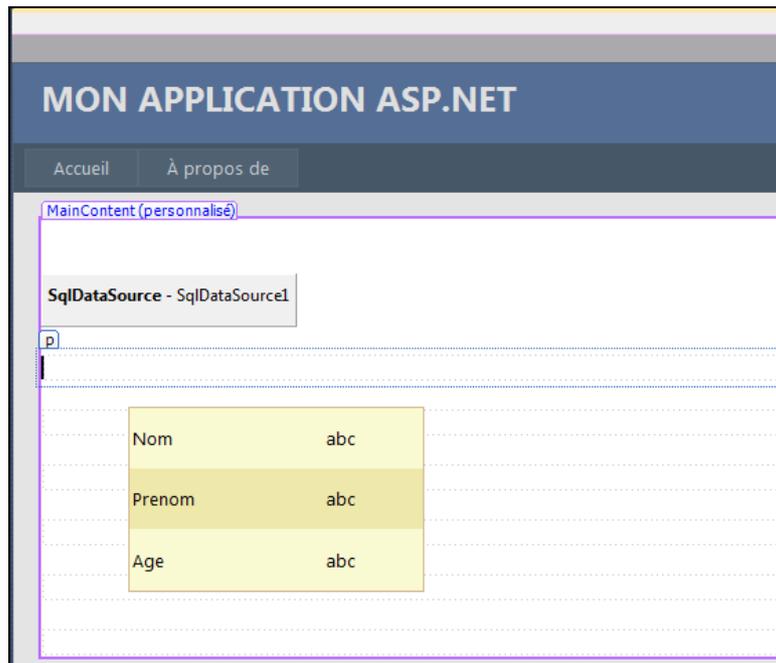
Ajouter un nouveau formulaire nommé **UseDetailView**.



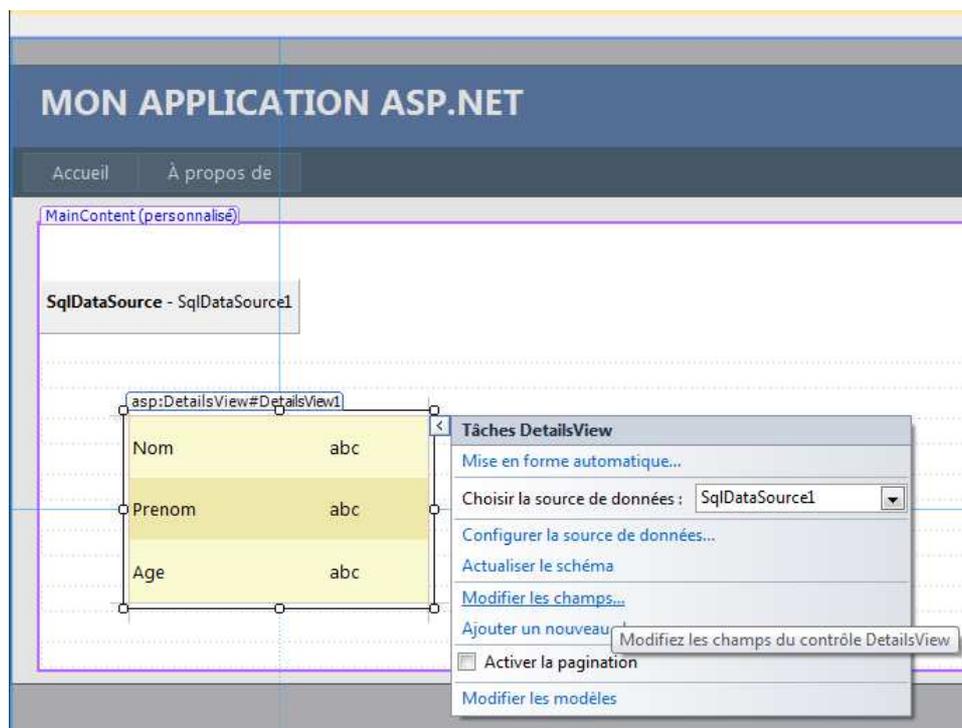
Posez ensuite un contrôle de type **DetailView** sur la page.



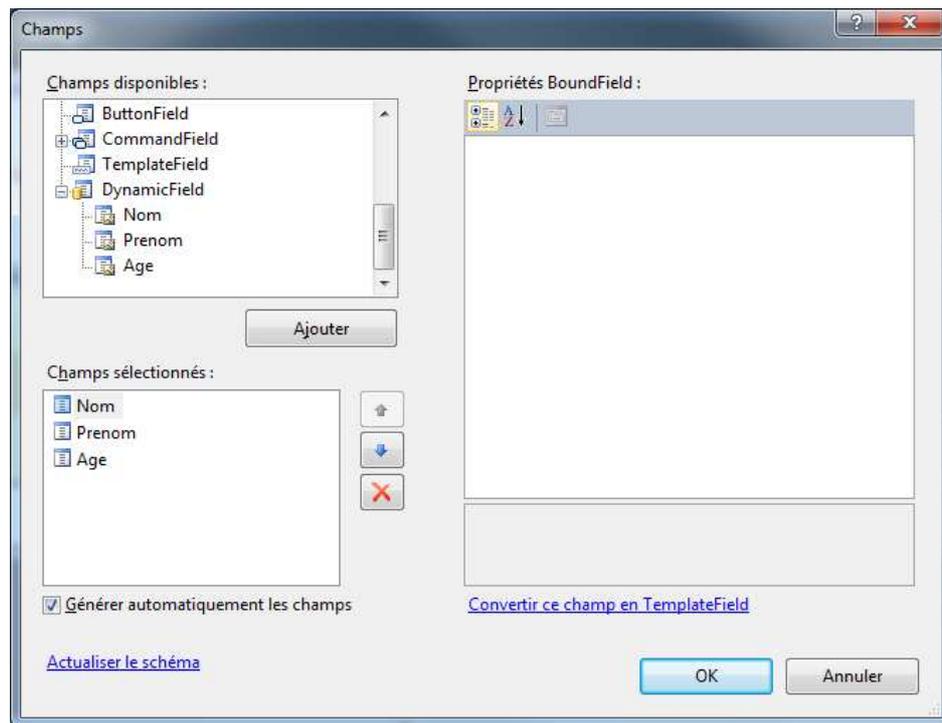
Procédez comme pour les deux cas précédents afin d'obtenir quelque chose qui ressemble à ceci :



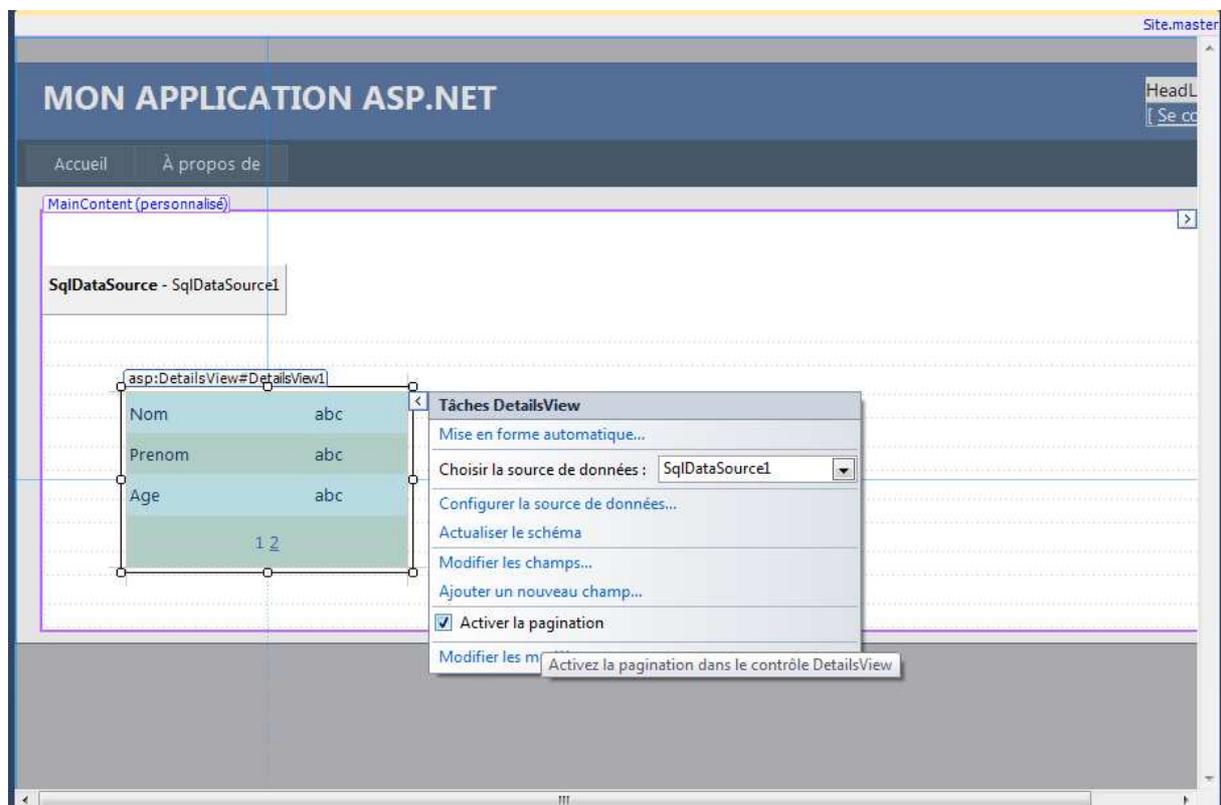
Faire ensuite **Modifier les champs**.



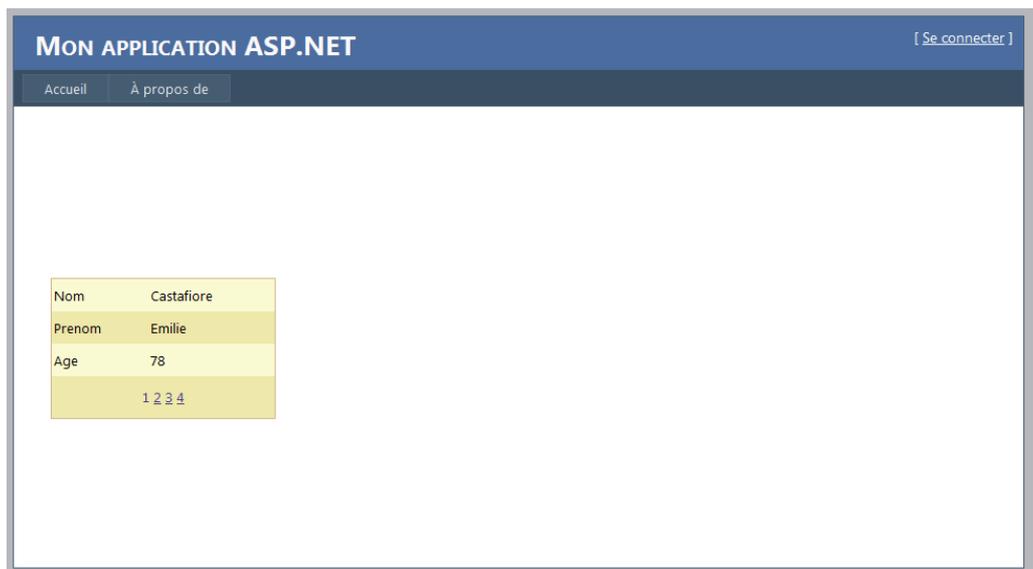
Ceci donne accès au menu suivant :



Activez ensuite la pagination.

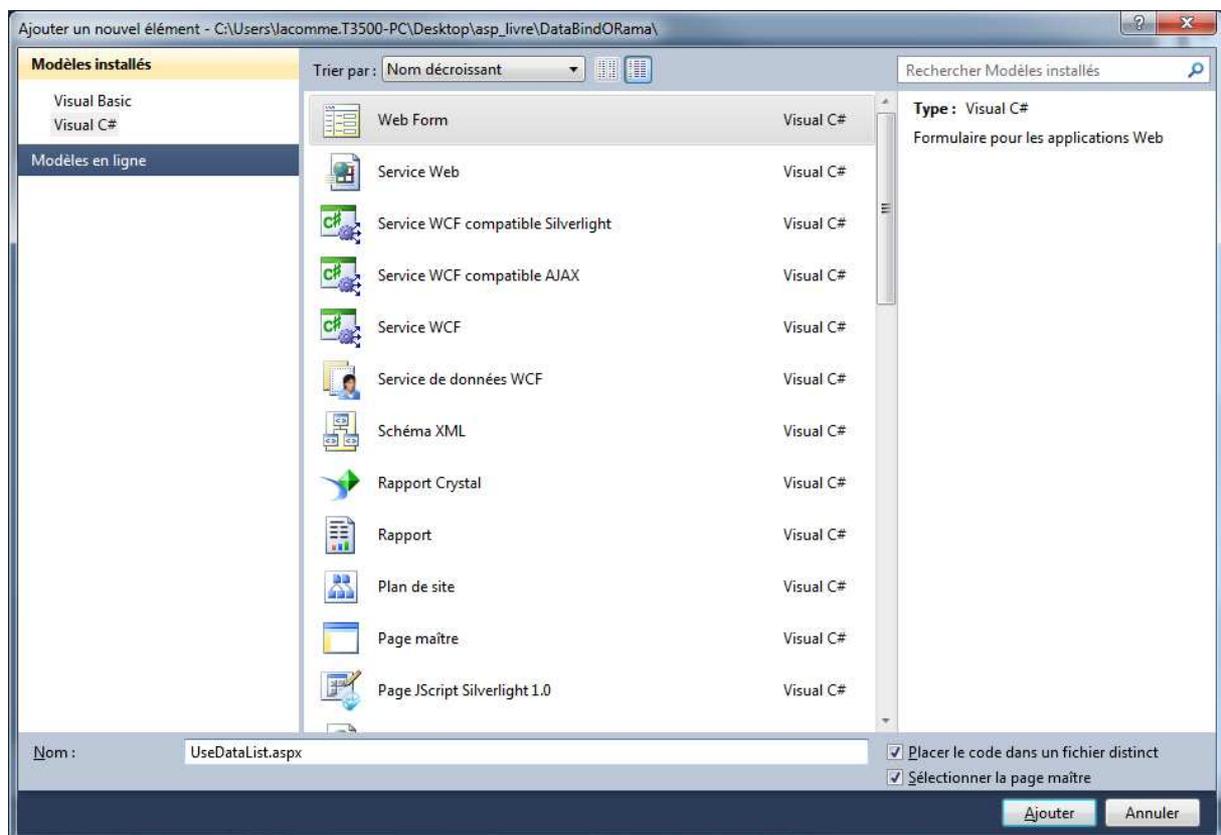


Le résultat d'exécution est le suivant :

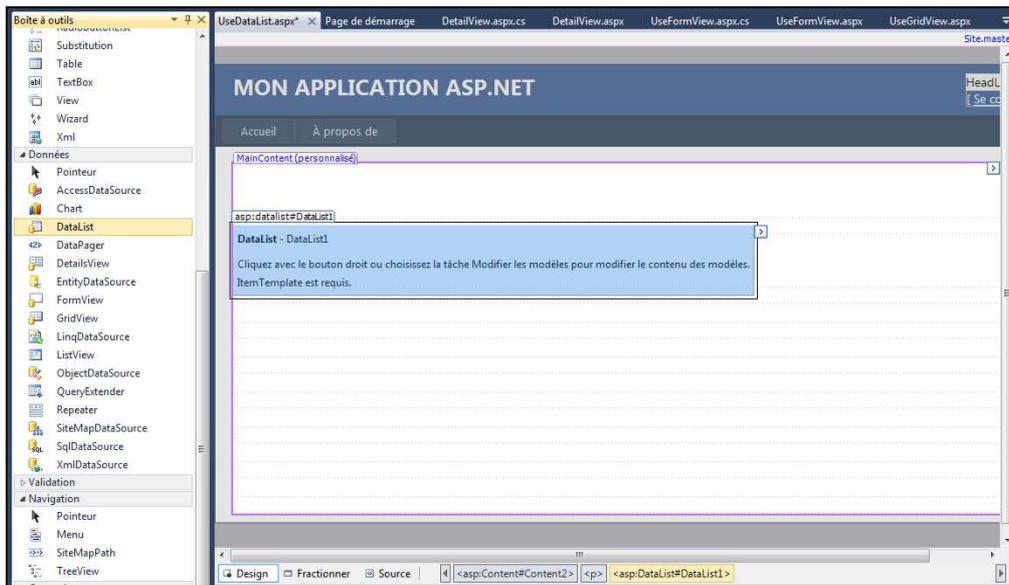


3.9. Utilisation des contrôles « DataList ».

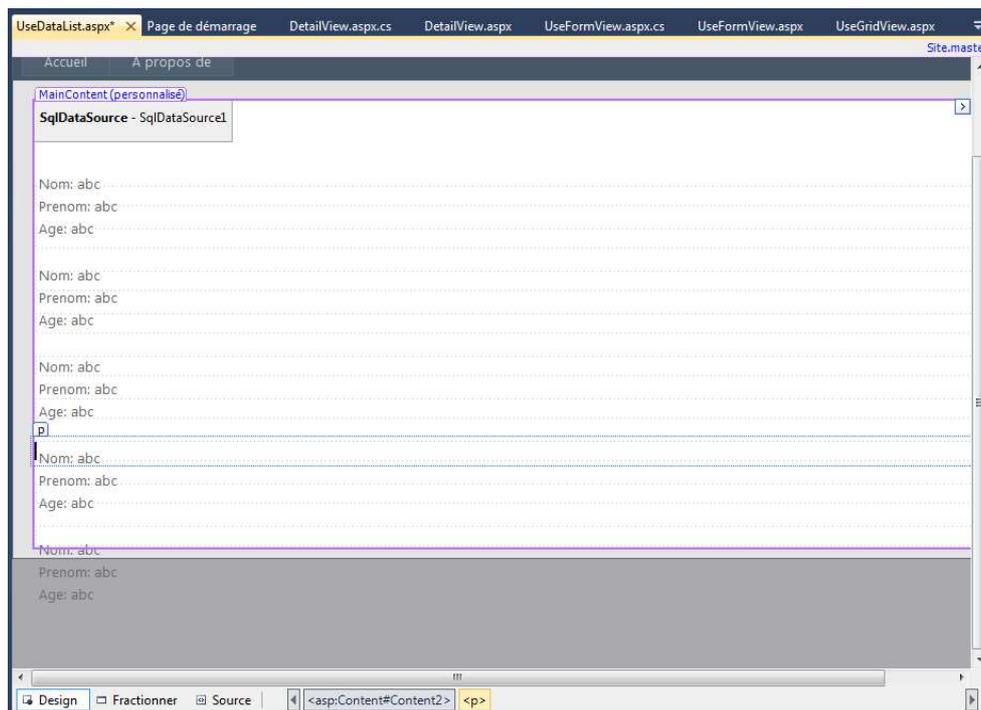
Ajouter un nouveau formulaire nommé **UseDataList**.



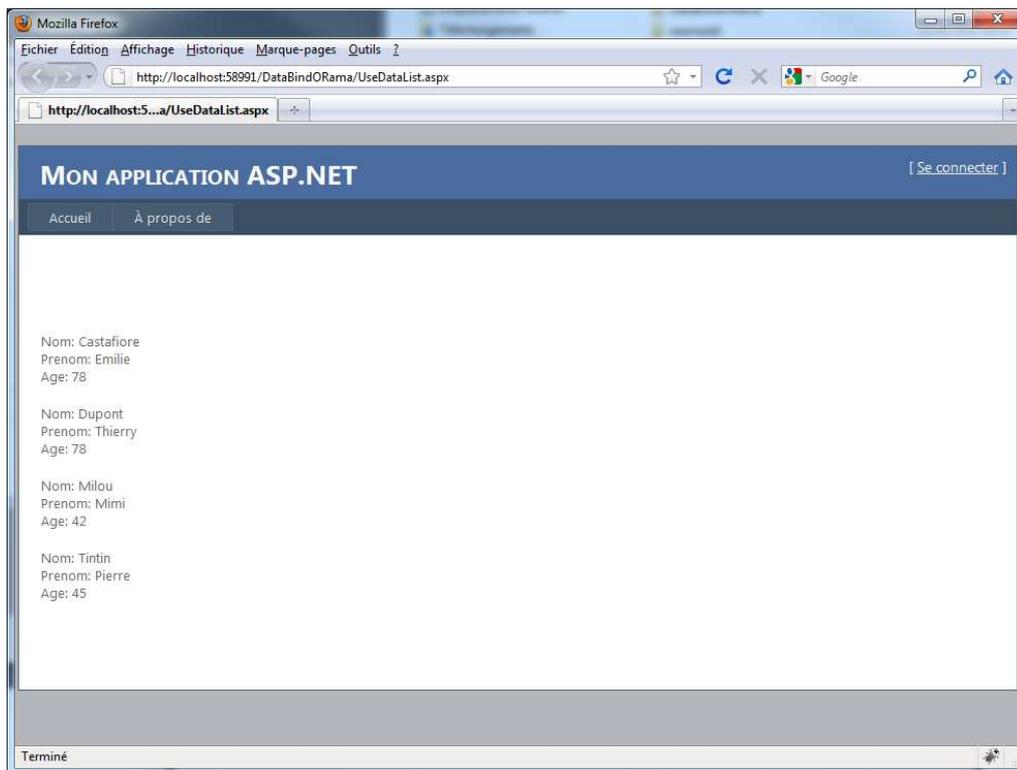
Posez ensuite un contrôle de type **Datalist** sur la page.



Procédez comme pour les deux cas précédents afin d'obtenir quelque chose qui ressemble à ceci :



Le résultat d'exécution est le suivant :



4. LINQ (Language Integrated Query)

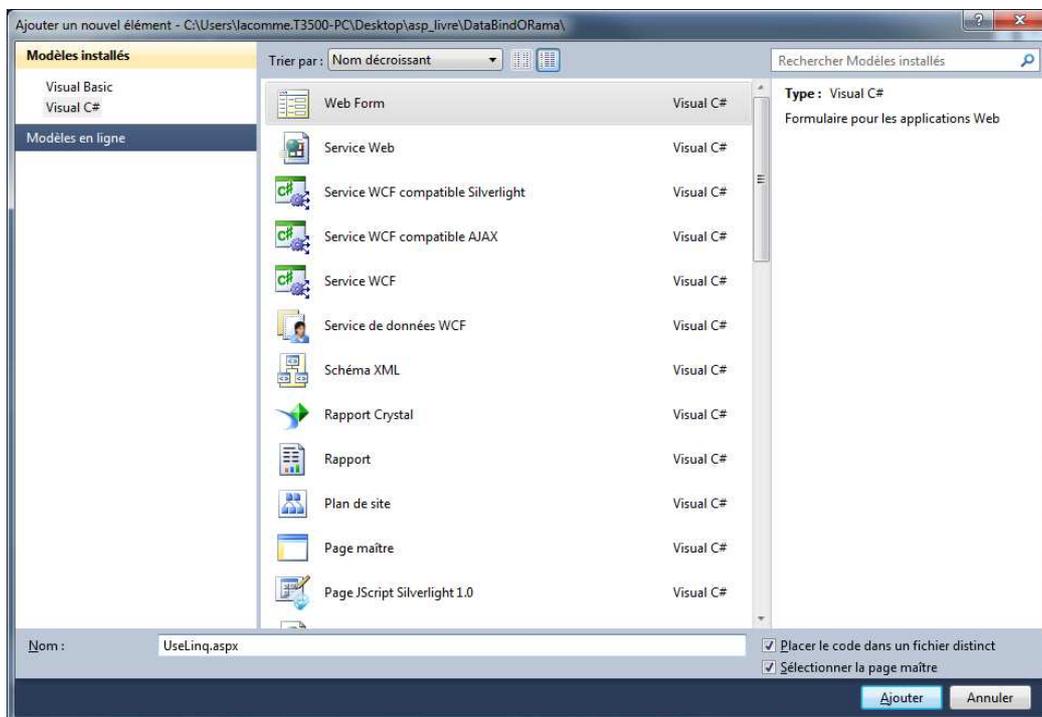


Objectifs :

- Présenter LINQ.

« *Language Integrated Query* (Requête intégrée au langage, aussi connu sous le nom de **LINQ**) est un composant du [Framework .NET](#) de [Microsoft](#) qui ajoute des capacités d'interrogation sur des données aux langages .NET en utilisant une syntaxe proche de celle de [SQL](#). » (Source Wikipedia : http://fr.wikipedia.org/wiki/Language_Integrated_Query).

Ajouter un nouveau formulaire nommé **UseLinq**.



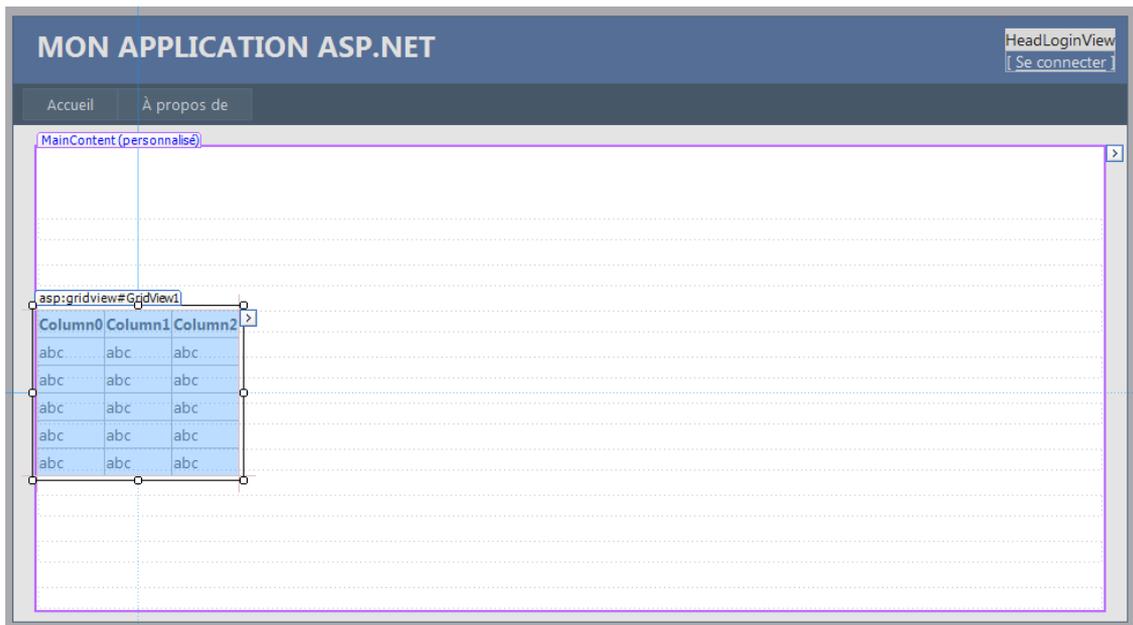
Le code de la page **UseLinq.aspx.cs** est le suivant :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class UseLinq : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
}
```

Déposer un **GridView** sur la page :



Nous allons relier le GridView aux éléments stockés dans une instance de la classe TechnologyDescriptor.

Modifiez le code **UseLinq.aspx.cs** comme suit :

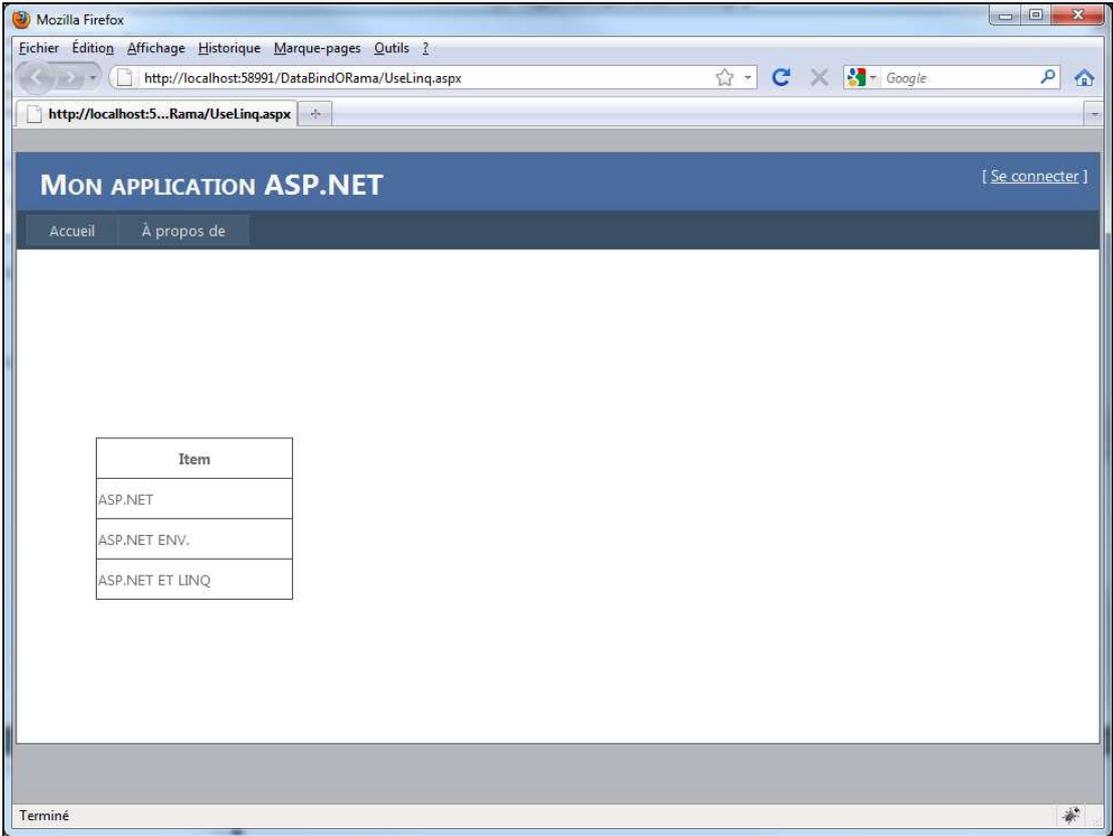
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class UseLinq : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        List<TechnologyDescriptor> techlist = TechnologyDescriptor.CreateTechnologyList();

        GridView1.DataSource = from ma_technologyDrescriptor in techlist
                               where ma_technologyDrescriptor.TechnologyName.Contains("NET") == true
                               orderby ma_technologyDrescriptor.TechnologyName
                               select ma_technologyDrescriptor.TechnologyName.ToUpper();

        GridView1.DataBind();
    }
}
```

Ceci donne à l'exécution :



5. Navigation dans un site web

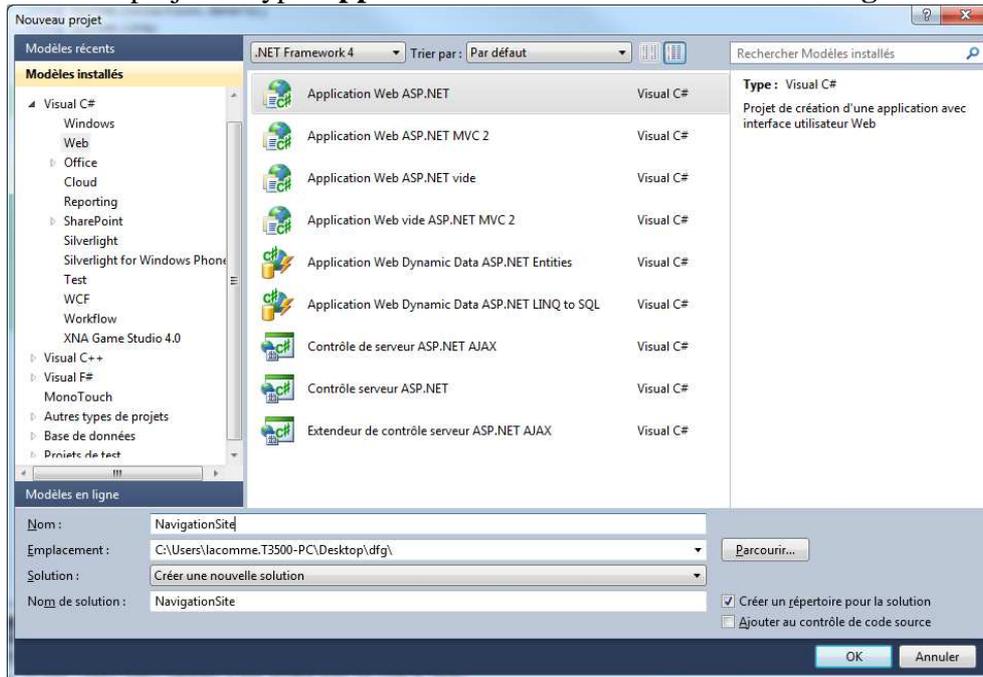


Objectifs :

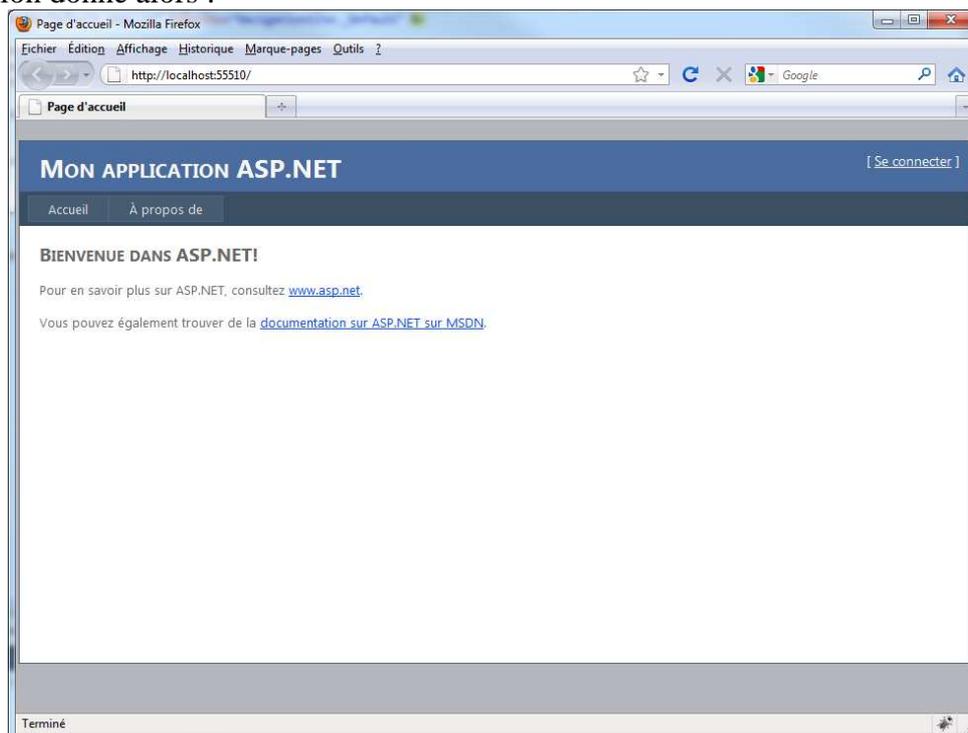
- comprendre les concepts de navigation sur un site
- utiliser le plan d'un site pour mettre en œuvre les contrôles de navigation
- gérer les événements.

5.1. Création d'un nouveau projet

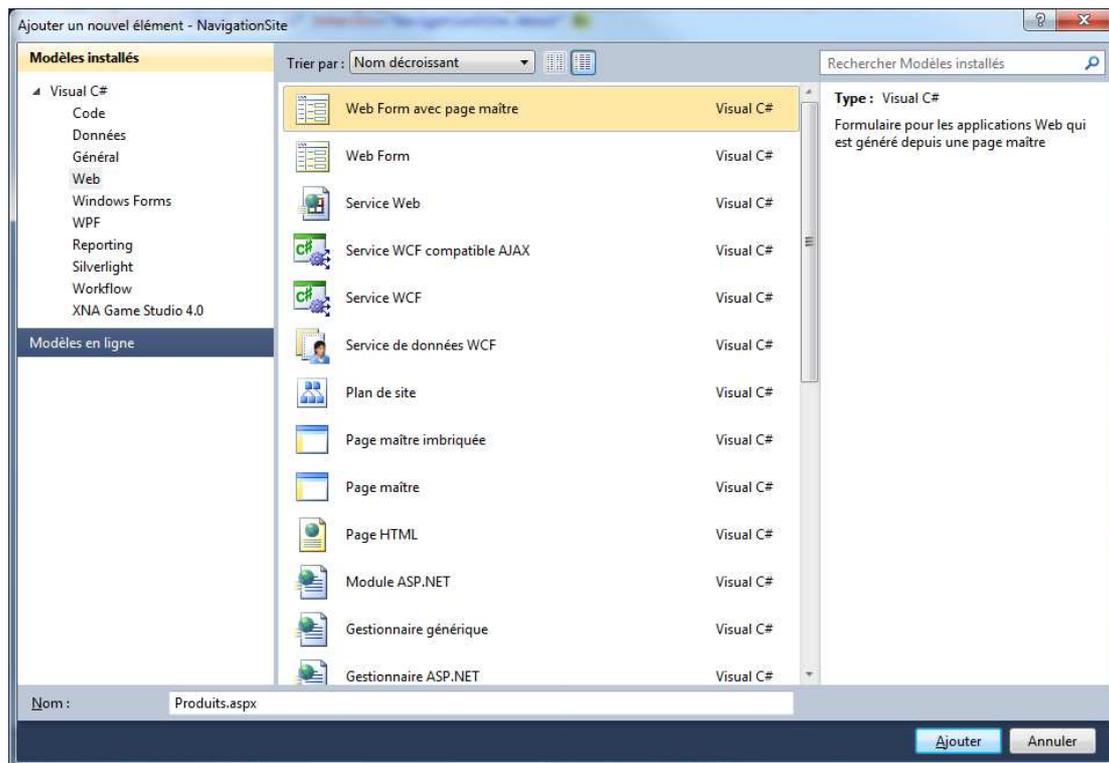
Créer un nouveau projet de type **ApplicationWeb ASP.NET** nommé **NavigationSite**.



L'exécution donne alors :



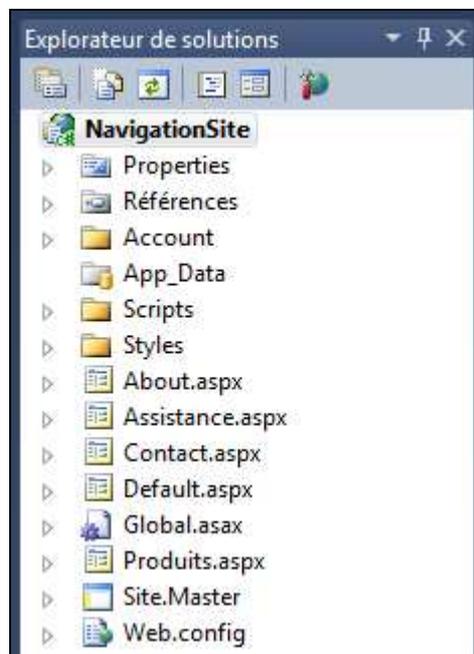
Ajoutez une page nommée **Produit.aspx**.



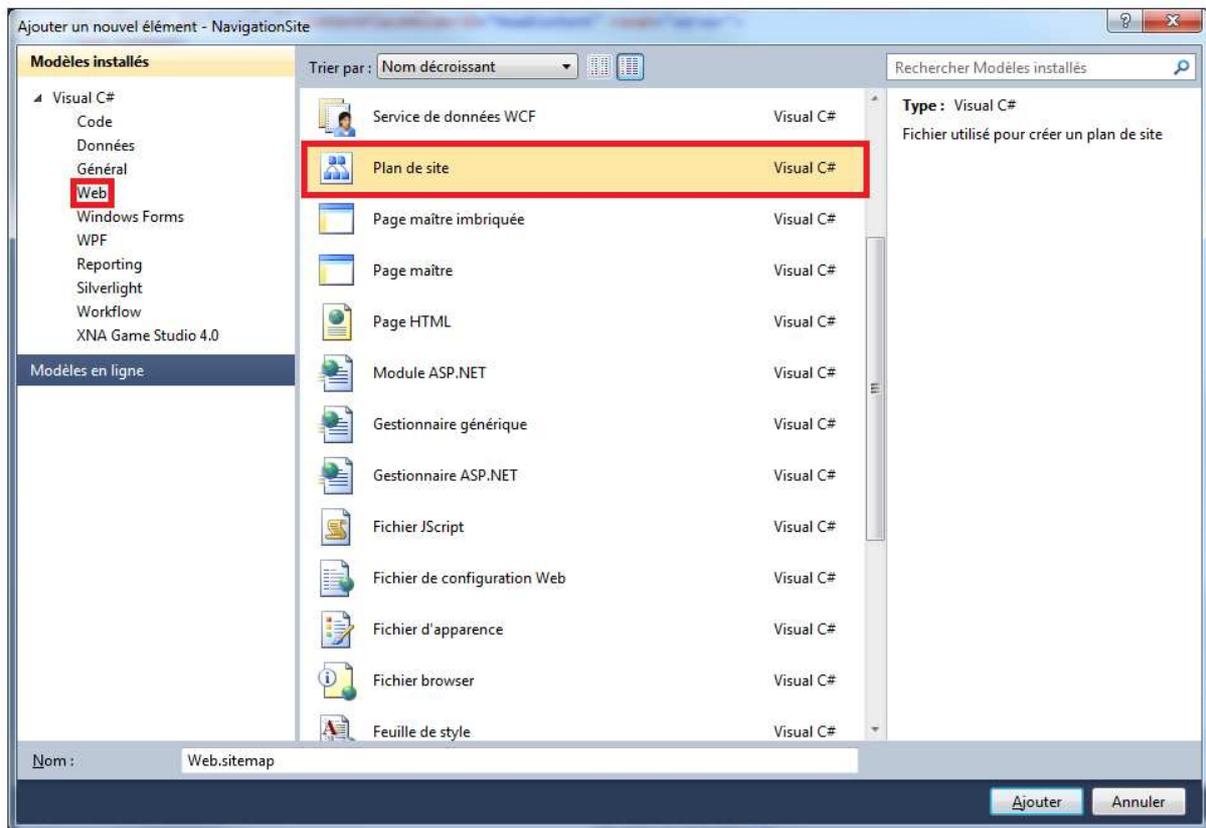
Ajoutez une page nommée **Assistance.aspx**.

Ajoutez une page nommée **Contact.aspx**.

Le projet ressemble à ceci :



Ajouter ensuite un plan de site comme suit :



Cela entraîne la création d'un fichier xml de la forme suivante :

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
  <siteMapNode url="" title="" description="">
    <siteMapNode url="" title="" description="" />
    <siteMapNode url="" title="" description="" />
  </siteMapNode>
</siteMap>
```

Modifier ce fichier de sorte qu'il ressemble à ce qui suit :

```
Site.Master.cs x Site.Master Web.sitemap x Contact.aspx Assistance.aspx Produits.aspx About.aspx Default.aspx
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
  <siteMapNode url="" title="Menu Principal" description="Menu Principal">
    <siteMapNode url="~/Produits.aspx" title="Gestion des produits en stock" description="Gestion des produits en stock" />
    <siteMapNode url="~/Contact.aspx" title="Les personnes à votre service" description="Les personnes à votre service" />
    <siteMapNode url="~/Assistance.aspx" title="Assistance téléphonique" description="Le service client" />
    <siteMapNode url="~/About.aspx" title="A propos de" description="A propos" />
  </siteMapNode>
</siteMap>
```

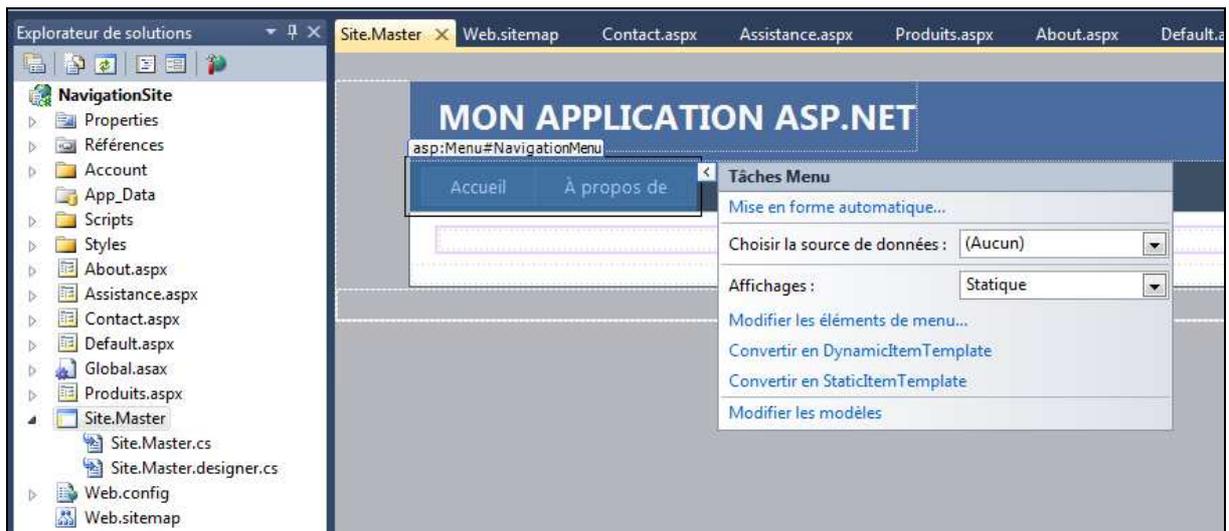
```

<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
  <siteMapNode url="" title="Menu Principal" description="Menu Principal">
    <siteMapNode url="~/Produits.aspx" title="Gestion des produits en stock" description="Gestion
des produits en stock" />
    <siteMapNode url="~/Contact.aspx" title="Les personnes à votre service" description="Les
personnes à votre service" />
    <siteMapNode url="~/About.aspx" title="A propos de" description="A propos" />
  </siteMapNode>
</siteMap>

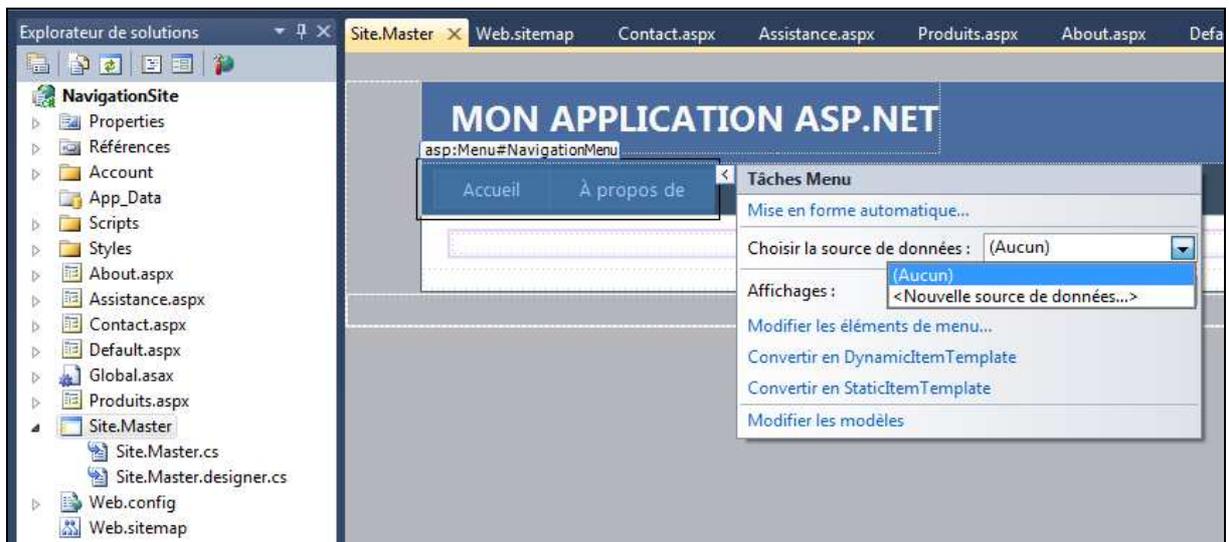
```

5.2. Ajout du plan de site

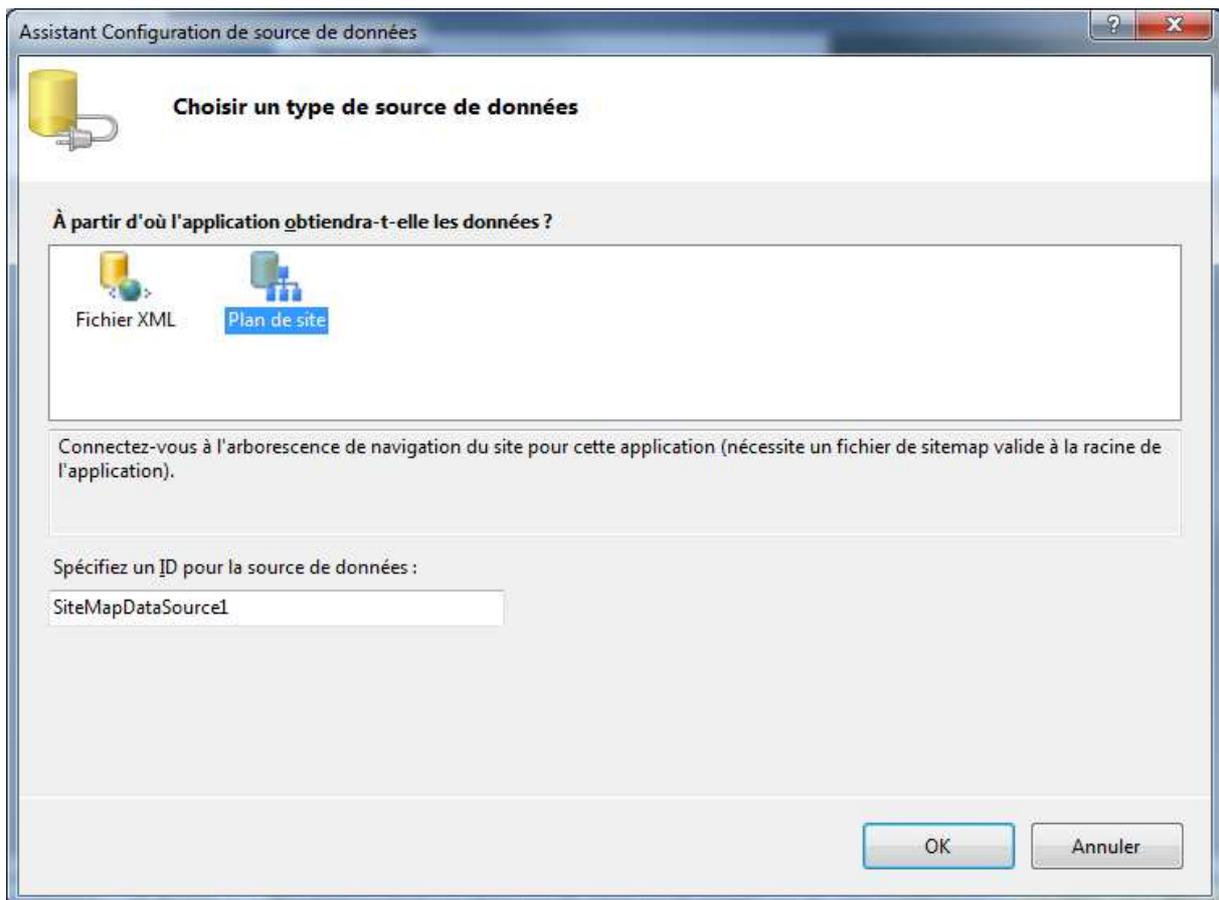
Ouvrir le fichier **site.master**.



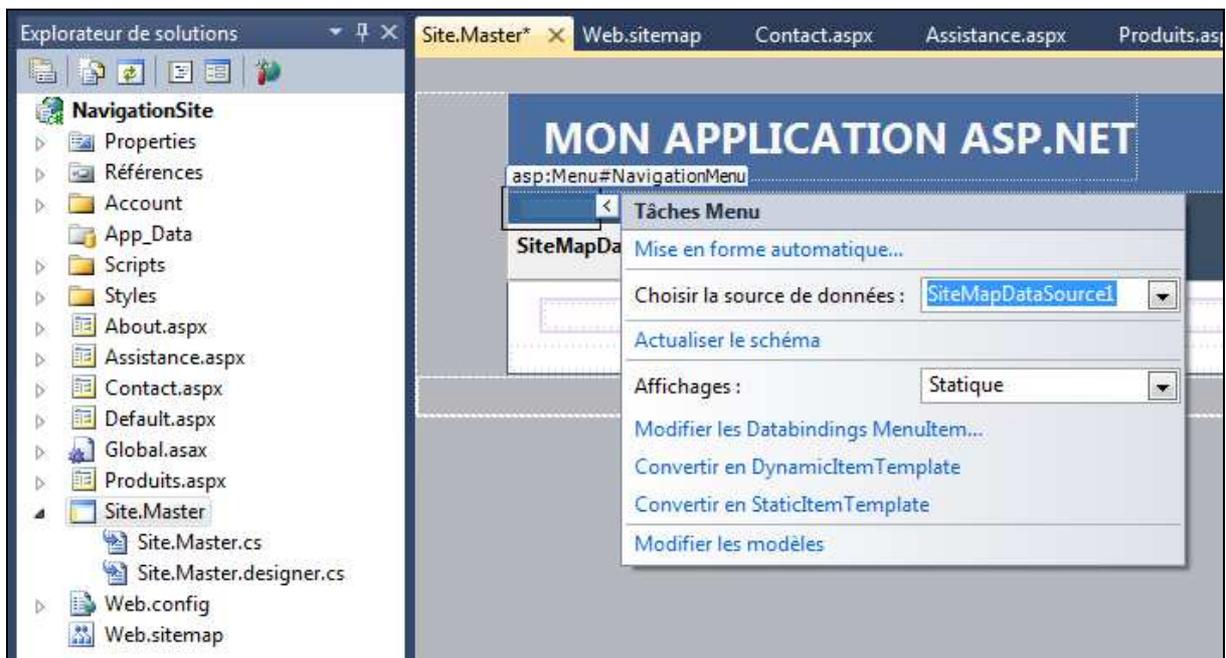
Choisir ensuite **Nouvelle Source de Données**.



Sélectionnez **Plan de Site**.



Ceci donne finalement :



A l'exécution on obtient un menu en haut de l'application.

