

Accès aux données

Contrôles liés aux données

- 3 types de contrôles liés aux données:
 - Listes
 - Tabulaires
 - Hiérarchiques
 - Graphiques

Les listes

- Classe de base: ListControl
- Types:
 - BulletedList: affiche une liste à puces, chaque élément peut être de type text, linkbutton ou hyperlink
 - CheckBoxList: affiche une liste de cases à cocher
 - DropDownList: affiche une liste déroulante
 - ListBox: affiche une liste d'éléments
 - RadioButtonList: affiche des boutons d'options
- Les 5 contrôles héritent de la classe de base ListControl.

Les listes

BulletedList

- Titanic
- Star Wars
- Jurassic Park
- Jaws
- Ghost
- Forrest Gump
- Ice Age
- Shrek
- Independence Day
- The Ring

CheckBoxList

- Titanic
- Star Wars
- Jurassic Park
- Jaws
- Ghost
- Forrest Gump
- Ice Age
- Shrek
- Independence Day
- The Ring

DropDownList

Titanic ▼

ListBox

Titanic
Star Wars
Jurassic Park
Jaws

RadioButtonList

- Titanic
- Star Wars
- Jurassic Park
- Jaws
- Ghost
- Forrest Gump
- Ice Age
- Shrek
- Independence Day
- The Ring

Les listes

```
<form id="form1" runat="server">
  <asp:BulletedList
    id="BulletedList1"
    DataSourceId="sdsFilms"
    Runat="server" DataTextField="titre" DataValueField="titre" />
  <asp:CheckBoxList
    id="CheckBoxList1"
    DataSourceId="sdsFilms"
    Runat="server" DataTextField="titre" />
  <asp:DropDownList
    id="DropDownList1"
    DataSourceId="sdsFilms"
    Runat="server" DataTextField="titre" />
  <asp:ListBox
    id="ListBox1"
    DataSourceId="sdsFilms"
    Runat="server" DataTextField="titre" />
  <asp:RadioButtonList
    id="RadioButtonList1"
    DataSourceId="sdsFilms"
    Runat="server" DataTextField="titre" />
  <asp:SqlDataSource
    id="sdsFilms"
    ConnectionString="<%= $ConnectionStrings:filmsConnectionString %>"
    SelectCommand="SELECT titre FROM films"
    Runat="server" />
</form>
```

Les listes

- Définition de la chaîne de connexion dans le fichier web.config.

```
<connectionStrings>
```

```
  <add name="filmsConnectionString" connectionString="
```

```
Data Source=DELL\SQLEXPRESS;
```

```
Initial Catalog=films;
```

```
Integrated Security=True"
```

```
  providerName="System.Data.SqlClient" />
```

```
</connectionStrings>
```

- La base de données peut être aussi attachée à partir du dossier App_Data:

```
<asp:SqlDataSource
```

```
  id="sdsFilms"
```

```
  ConnectionString="Data Source=.\SQLEXPRESS;
```

```
  AttachDbFilename=|DataDirectory|films.mdf;
```

```
  Integrated Security=True;User Instance=True"
```

```
  SelectCommand="SELECT Titre FROM films"
```

```
  Runat="server" />
```

Contrôles tabulaires liés aux données

- ASP .NET offre six contrôles tabulaires qui peuvent être classés en deux catégories:
 - Les contrôles qui permettent l'affichage de plusieurs enregistrements en une seule fois.
 - GridView: affiche un ensemble d'enregistrements dans une table html, et supporte le tri, la sélection, la pagination, la suppression et la modification des données.
 - DataList: affiche des données dans une table html, plusieurs enregistrements peuvent être affichés dans une même ligne.
 - Repeater: affiche des données en utilisant un modèle
 - ListView: affiche des données en utilisant un modèle, et supporte la pagination, la sélection et la modification des données
 - Les contrôles qui affichent un enregistrement à la fois.
 - DetailsView: affiche un enregistrement dans une table html, ce contrôle supporte la pagination, l'ajout et la modification des données.
 - FormView: utilise un modèle.

Contrôles tabulaires liés aux données

GridView

titre	Réalisateur
Titanic	James Cameron
Star Wars	George Lucas
Jurassic Park	Steven Spielberg
Jaws	Steven Spielberg
Ghost	Jerry Zucker
Forrest Gump	Robert Zemeckis
Ice Age	Chris Wedge
Shrek	Andrew Adamson
Independence Day	Roland Emmerich
The Ring	Gore Verbinski

DataList

Titanic	Forrest Gump
Réalisateur: James Cameron	Réalisateur: Robert Zemeckis
Star Wars	Ice Age
Réalisateur: George Lucas	Réalisateur: Chris Wedge
Jurassic Park	Shrek
Réalisateur: Steven Spielberg	Réalisateur: Andrew Adamson
Jaws	Independence Day
Réalisateur: Steven Spielberg	Réalisateur: Roland Emmerich
Ghost	The Ring
Réalisateur: Jerry Zucker	Réalisateur: Gore Verbinski

DetailsView

titre	Titanic
Réalisateur	James Cameron
1 2 3 4 5 6 7 8 9 10	

FormView

titre: Titanic
Réalisateur: James Cameron
1 2 3 4 5 6 7 8 9 10

Repeater

Titanic Réalisé par James Cameron Star Wars Réalisé par George Lucas Jurassic Park Réalisé par Steven Spielberg Jaws Réalisé par Steven Spielberg Ghost Réalisé par Jerry Zucker Forrest Gump Réalisé par Robert Zemeckis Ice Age Réalisé par Chris Wedge Shrek Réalisé par Andrew Adamson Independence Day Réalisé par Roland Emmerich The Ring Réalisé par Gore Verbinski

ListView

Titanic directed by James Cameron Star Wars directed by George Lucas Jurassic Park directed by Steven Spielberg 1 [2](#) [3](#) [4](#)

GridView

- Un GridView est constitué d'une collection (Rows) de type GridViewRow et d'une collection (Columns) de type DataControlField
- GridViewRow dérive de TableRow qui contient la propriété Cells (une collection de type DataControlFieldCell)
- DataControlField implémente la méthode InitializeCell

Styles

- GridViewStyle
- HeaderStyle,
- FooterStyle,
- RowStyle,
- AlternatingRowStyle,
- SelectedRowStyle,
- EditRowStyle

Les styles peuvent être définis d'une manière déclarative ou bien dans le code à l'aide des événements RowCreated et RowDataBound, le style peut être appliqué au niveau de chaque cellule. RowCreated est déclenché avant le chargement des données, alors que RowDataBound survient après le chargement des données.

Les deux événements RowCreated et RowDataBound surviennent après application des styles.

Contrôles hiérarchiques liés aux données

- Menu:
- TreeView

Contrôles hiérarchiques liés aux données

TreeView

- [-] Fiction
 - StarWars
 - JurassicPark
 - IndependenceDay
- [-] Animation
 - IceAge
 - Shrek
- [-] Drame
 - Titanic
 - Ghost
 - ForrestGump
- [-] Horreur
 - Saw5
 - Jaws

Menu

- Fiction ▶
- Animation ▶
- Drame ▶
- Horreur ▶ Saw5
- Jaws

Contrôles hiérarchiques liés aux données

```
<form id="form1" runat="server">  
  <asp:TreeView  
    id="CheckBoxList1"  
    DataSourceId="xdsFilms"  
    Runat="server" />  
  <asp:Menu  
    id="BulletedList1"  
    DataSourceId="xdsFilms"  
    Runat="server" />  
  <asp:XmlDataSource  
    id="xdsFilms"  
    DataFile="~/Films.xml"  
    XPath="/Films/*"  
    Runat="server" />  
</form>
```

Les contrôles DataSource

- Contrôles de type DataSourceControl: contrôle qui représente les données sous une forme tabulaire
 - SqlDataSource: données extraites à partir d'une base de données relationnelle: Microsoft SQL Server, Oracle, DB2.
 - LinqDataSource: peut être associé à une requête LINQ TO SQL
 - AccessDataSource: peut être associé à une base de données Microsoft Access
 - ObjectDataSource: données extraites à partir d'un objet métier
 - EntityDataSource: données extraites à partir d'une entité du modèle EDM.
- Contrôles de type HierarchicalDataSourceControl: contrôle qui permet une représentation des données à la fois tabulaires et hiérarchiques
 - XmlDataSource: la source de données doit être un document XML
 - SiteMapDataSource: données extraites à partir d'un fournisseur Site Map, un SiteMap représente la structure d'un site web

GridView

```
<asp:GridView
  id="GridView1"
  DataSourceId="sdsFilms"
  Runat="server" AutoGenerateColumns="False" DataKeyNames="Id" >
  <Columns>
    <asp:BoundField DataField="Id" HeaderText="Id" InsertVisible="False"
      ReadOnly="True" SortExpression="Id" />
    <asp:BoundField DataField="catid" HeaderText="catid" SortExpression="catid" />
    <asp:BoundField DataField="titre" HeaderText="titre" SortExpression="titre" />
    <asp:BoundField DataField="Réalisateur" HeaderText="Réalisateur"
      SortExpression="Réalisateur" />
    <asp:BoundField DataField="date" HeaderText="date" SortExpression="date" />
  </Columns>
</asp:GridView>
```

```
<asp:SqlDataSource
  id="sdsFilms"
  ConnectionString="<%= $ConnectionStrings:filmsConnectionString %>"
  SelectCommand="SELECT Id, catid, titre, Réalisateur, date FROM films"
  Runat="server" />
```

Id	catid	titre	Réalisateur	date
1	3	Titanic	James Cameron	21.06.1997 00:00:00
2	1	Star Wars	George Lucas	01.06.1977 00:00:00
3	1	Jurassic Park	Steven Spielberg	17.06.1993 00:00:00
4	4	Jaws	Steven Spielberg	30.05.1975 00:00:00
5	3	Ghost	Jerry Zucker	14.06.1990 00:00:00
7	3	Forrest Gump	Robert Zemeckis	18.06.1994 00:00:00
8	2	Ice Age	Chris Wedge	26.06.2002 00:00:00
9	2	Shrek	Andrew Adamson	25.06.2001 00:00:00
10	1	Independence Day	Roland Emmerich	20.06.1996 00:00:00
11	4	The Ring	Gore Verbinski	26.06.2002 00:00:00

GridView

```
<asp:GridView
  id="GridView1"
  DataSourceId="sdsFilms"
  Runat="server" AutoGenerateColumns="False" DataKeyNames="Id"
  AllowPaging="True" AllowSorting="True" >
  <Columns>
    <asp:CommandField ShowSelectButton="True" />
    <asp:BoundField DataField="Id" HeaderText="Id" InsertVisible="False"
      ReadOnly="True" SortExpression="Id" />
    <asp:BoundField DataField="catid" HeaderText="catid" SortExpression="catid" />
    <asp:BoundField DataField="titre" HeaderText="titre" SortExpression="titre" />
    <asp:BoundField DataField="Réalisateur" HeaderText="Réalisateur"
      SortExpression="Réalisateur" />
    <asp:BoundField DataField="date" HeaderText="date" SortExpression="date" />
  </Columns>
</asp:GridView>
```

```
<asp:SqlDataSource
  id="sdsFilms"
  ConnectionString="<%$ ConnectionStrings.filmsConnectionString %>"
  SelectCommand="SELECT Id, catid, titre, Réalisateur, date FROM films"
  Runat="server" />
```

```
protected void Menu1_MenuItemClick(object sender, MenuEventArgs e)
{Label1.Text = Menu1.Selected.Value; }
protected void TreeView1_SelectedNodeChanged(object sender, EventArgs e)
{Label2.Text = TreeView1.Selected.Value;}
```

Les paramètres ASP.NET

- Les contrôles SqlDataSource, AccessDataSource, LinqDataSource, et ObjectDataSource supportent les types de paramètres suivants:
 - Parameter: représente une valeur statique arbitraire.
 - ControlParameter: représente la valeur d'une propriété d'une page ou d'un contrôle.
 - CookieParameter: représente la valeur d'un cookie.
 - FormParameter: représente la valeur d'un champ de formulaire envoyé par la méthode POST.
 - ProfileParameter: représente la valeur d'une propriété de profile.
 - QueryStringParameter: représente la valeur d'un champ de formulaire envoyé par la méthode GET
 - SessionParameter: représente la valeur d'un élément stocké dans une variable de session.

Exemple: ControlParameter

```
<asp:DropDownList
  id="ddlCategories"
  DataSourceID="sdsFilmsCategories"
  DataTextField="nom"
  DataValueField="Id"
  Runat="server" />
<asp:Button
  id="btnSelect"
  Text="Select"
  ToolTip="Sélectionner un film"
  Runat="server" />
<hr />
<asp:GridView
  id="grdFilms"
  DataSourceID="sdsFilms"
  Runat="server" AutoGenerateColumns="False" >
  <Columns>
    <asp:BoundField DataField="Réalisateur" HeaderText="Réalisateur"
      SortExpression="Réalisateur" />
```

Horreur ▼ Select

Réalisateur	titre
Steven Spielberg	Jaws
Gore Verbinski	The Ring

Exemple: ControlParameter

```
<asp:BoundField DataField="titre" HeaderText="titre" SortExpression="titre" />
```

```
<asp:BoundField DataField="catid" Visible="false" HeaderText="catid" SortExpression="catid" />
```

```
</Columns>
```

```
</asp:GridView>
```

```
<asp:SqlDataSource
```

```
  id="sdsFilmsCategories"
```

```
  ConnectionString="<%= $ConnectionStrings.filmsConnectionString %>"
```

```
  SelectCommand="SELECT * FROM [categories]"
```

```
  Runat="server" />
```

```
<asp:SqlDataSource
```

```
  id="sdsFilms"
```

```
  ConnectionString="<%= $ConnectionStrings.filmsConnectionString %>"
```

```
  SelectCommand="SELECT Réalisateur, titre, catid FROM films WHERE (catid = @idr)"
```

```
  Runat="server">
```

```
  <SelectParameters>
```

```
    <asp:ControlParameter
```

```
      Name="idr"
```

```
      Type="Int32"
```

```
      ControlID="ddlCategories" PropertyName="SelectedValue" />
```

```
  </SelectParameters>
```

```
</asp:SqlDataSource>
```

Lier un contrôle par programme

- Un contrôle lié aux données possède une propriété DataSource qui définit la source de données et une méthode DataBind() qui lie le contrôle à la source de données.
- Exemple: afficher la liste des polices installées dans une grille
using System.Drawing.Text;

...

```
protected void Page_Load(object sender,
    EventArgs e)
{
    if (!IsPostBack)
    {
        InstalledFontCollection f = new
        InstalledFontCollection();
        g1.DataSource = f.Families;
        g1.DataBind();
    }
}
```

Name
Agency FB
Aharoni
Algerian
Andalus
Angsana New
AngsanaUPC
Arabic Typesetting
Arial

Remarques

- La propriété DataSource peut contenir un objet qui implémente l'une des interfaces suivantes: IEnumerable ou IList
- Une exception est déclenchée si on définit les deux propriétés DataSource et DataSourceId
- La méthode DataBind() peut être appelée même si la propriété DataSourceId est définie, dans ce cas là le contenu de la grille sera rafraîchi

Templates et expressions

DataBindings

- Tous les contrôles de liaison aux données supportent les templates à l'exception du contrôle TreeView
- L'utilisation des templates est obligatoire dans le cas des contrôles Repeater, ListView et FormView.
- Un modèle peut contenir des éléments html, des expressions databings et d'autres contrôles, Exemple:
- `<ItemTemplate>`

```
<asp:HyperLink
  id="HyperLink1"
  Text='<%# Eval("titre") %>'
  NavigateUrl='<%# Eval("Id", "Details.aspx?id={0}") %>'
  runat="server" />
<br />
</ItemTemplate>
```

Exemple : Repeater

```
<asp:Repeater
  id="Repeater1"
  DataSourceId="sdsFilms"
  Runat="server" onitemcommand="Repeater1_ItemCommand">
  <ItemTemplate>
    <%#Eval("titre")%>
    <i>Réalisé par</i>
    <%#Eval("réalisateur")%>
    <hr />
  </ItemTemplate>
</asp:Repeater>
```

```
<asp:SqlDataSource
  id="sdsFilms"
  ConnectionString="<%$
  ConnectionStrings:filmsConnectionString %>"
  SelectCommand="SELECT titre, Réalisateur FROM films"
  Runat="server" />
```

Titanic Réalisé par James Cameron

Star Wars Réalisé par George Lucas

Jurassic Park Réalisé par Steven Spielberg

Jaws Réalisé par Steven Spielberg

DetailsView

- Exemple:

```
<asp:DetailsView
  id="dvFilms"
  DataSourceID="srcFilms"
  Runat="server" AutoGenerateRows="False" DataKeyNames="Id" >
  <Fields>
    <asp:BoundField DataField="Id" HeaderText="Id" InsertVisible="False"
      ReadOnly="True" SortExpression="Id" />
    <asp:BoundField DataField="titre" HeaderText="titre" SortExpression="titre" />
    <asp:BoundField DataField="Réalisateur" HeaderText="Réalisateur"
      SortExpression="Réalisateur" />
    <asp:CheckBoxField DataField="enSalle" HeaderText="enSalle"
      SortExpression="enSalle" />
  </Fields>
</asp:DetailsView>
```

Id	1
titre	Titanic
Réalisateur	James Cameron
enSalle	<input type="checkbox"/>

```
<asp:SqlDataSource
  id="srcFilms"
  ConnectionString="<%= $ConnectionStrings:films %>"
  SelectCommand="SELECT Id, titre, Réalisateur, enSalle FROM films WHERE (Id = 1)"
  Runat="server" />
```

DetailView

- **BoundField**: Afficher le champ dans une zone de texte.
- **CheckBoxField** :
- **CommandField**: afficher des commandes de modification, suppression et sélection.
- **ButtonField**: affiche la valeur d'un d'un champ comme un bouton (imagebutton, linkbutton, ou push button).
- **HyperLinkField**: affiche la valeur dans un lien
- **ImageField**—Afficher la valeur d'un champ comme une image.
- **TemplateField**—utiliser un template.

FormView

- FormView permet la définition des templates suivantes: ItemTemplate, EditItemTemplate, InsertItemTemplate, EmptyDataTemplate, FooterTemplate, HeaderTemplate, et PagerTemplate.

Repeater

- Supporte uniquement le template ItemTemplate.

List View

- Ce contrôle supporte plus de templates que les autres contrôles:
 - LayoutTemplate: définit le template dans lequel les lignes seront affichées
 - GroupTemplate: propriété GroupItemCount
 - ItemSeparatorTemplate

```
<asp:ListView ID="lstFilms"
  DataSourceID="sdsFilms"
  DataKeyNames="ID"
  runat="server">
  <LayoutTemplate>
    <table cellpadding="2" width="640px" border="1" runat="server"
id="tblCategories">
      <tr id="Tr1" runat="server">
        <th id="Th1" runat="server"></th>
        <th id="Th2" runat="server">ID</th>
        <th id="Th3" runat="server">Catégorie</th>
      </tr>
      <tr runat="server" id="itemPlaceholder" />
    </table>
```

```

<asp:DataPager runat="server" ID="CategoriesDataPager" PageSize="12">
    <Fields>
        <asp:NextPreviousPagerField ShowFirstPageButton="true"
ShowLastPageButton="true"
        FirstPageText="|&lt;&lt; " LastPageText=" &gt;&gt;|"
        NextPageText=" &gt; " PreviousPageText=" &lt; " />
    </Fields>
</asp:DataPager>
</LayoutTemplate>
<ItemTemplate>
    <tr id="Tr2" runat="server">
        <td>
            <asp:LinkButton ID="EditButton" runat="Server" Text="Modifier"
CommandName="Edit" />
        </td>
        <td>
            <asp:Label ID="lblId" runat="Server" Text='<#Eval("ID") %>'
/>
        </td>
        <td valign="top">
            <asp:Label ID="lblNom" runat="Server" Text='<#Eval("Nom") %>'
/></td></tr></ItemTemplate>

```

```
<EditItemTemplate>
    <tr style="background-color: #ADD8E6"><td>
        <asp:LinkButton ID="UpdateButton" runat="server"
CommandName="Update" Text="Modifier" />&nbsp;
        <asp:LinkButton ID="CancelButton" runat="server"
CommandName="Cancel" Text="Annuler" />
    </td>
    <td>
        <asp:TextBox ID="txtID" runat="server" Text='<#Bind("ID") %>'
ReadOnly="true"
        MaxLength="50" /><br />
    </td>
    <td>
        <asp:TextBox ID="txtNom" runat="server" Text='<#Bind("Nom")
%>'
        MaxLength="50" /><br />
    </td>
</tr>
</EditItemTemplate>
</asp:ListView>
```

Exemple 2: utilisation d'un contrôle DataPage et de ItemSeparatorTemplate

```
<asp:ListView runat="server"
ID="ListView1"
DataKeyNames="ID"
DataSourceID="sdsFilms">
<LayoutTemplate>
<div id="itemPlaceholder"
runat="server"></div>
<br />
<div style="text-align: center">
<asp:DataPager ID="DataPager1"
runat="server" PageSize="4">
<Fields>
<asp:NextPreviousPagerField
ButtonType="Button"
ShowFirstPageButton="True"
ShowLastPageButton="True" />
</Fields>
</asp:DataPager>
</div>
</LayoutTemplate>
<ItemTemplate>
<div style="text-align: center">
<b>Titre:</b>
<asp:Label ID="lblTitre" runat="server"
Text='<%# Eval("Titre") %>' />
<br />
<b>Réalisateur:</b>
<asp:Label ID="lblRealisateur"
runat="server"
Text='<%# Eval("Réalisateur") %>' />
<br />
<b>Date:</b>
<asp:Label ID="lblDate" runat="server"
Text='<%# Eval("Date") %>' />
<br />
</div>
</ItemTemplate>
<ItemSeparatorTemplate>
<hr />
</ItemSeparatorTemplate>
</asp:ListView>
```

ChartControl

- Le contrôle, génère une image (format png par défaut) et l'envoie dans la réponse http, lors de son exécution
- Le contrôle supporte 25 différentes types de graphiques.

```
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%$ ConnectionStrings:csBase %>"
SelectCommand="SELECT SUM([Order Details].UnitPrice * [Order
Details].Quantity) AS
Total, Orders.ShipCountry FROM Orders INNER JOIN [Order Details] ON
Orders.OrderID =
[Order Details].OrderID GROUP BY Orders.ShipCountry Order By
Total"></asp:SqlDataSource>
```

```
<asp:Chart ID="Chart1" runat="server" DataSourceID="SqlDataSource1"
Width="702px" Height="581px">
<Series>
<asp:Series Name="Series1" XValueMember="ShipCountry" YValueMembers="Total"
ChartType="Bar" XValueType="String" IsValueShownAsLabel="True"
LabelBackColor="White" LabelFormat="{c}" >
<SmartLabelStyle CalloutBackColor="White" />
</asp:Series>
</Series>
<ChartAreas>
<asp:ChartArea Name="ChartArea1">
<AxisY>
<LabelStyle Format="{c}" />
</AxisY>
<AxisX Interval="1">
</AxisX>
</asp:ChartArea>
</ChartAreas>
</asp:Chart>
```

DataList

- Similaire au contrôle Repeater, mais permet de définir la direction d'affichage des enregistrements: RepeatDirection.
- Le contrôle DataList n'utilise pas automatiquement un DataSource pour l'édition, il fournit plutôt des événements pour effectuer les modifications. pour activer une opération de modification, il faut ajouter un bouton et définir sa propriété CommandName (edit, delete, update, ou cancel)

TreeView

- Peut contenir une collection Nodes ou DataBindings
- **Fichier clients.xml**

```
<?xml version="1.0" encoding="utf-8" ?>
<Clients>
  <Client ClientId="1" Nom="Northwind Traders">
    <Commandes>
      <Commande CommandeId="1" LivraisonDate="06-22-2006">
        <CommandeLignes>
          <CommandeLigne CommandeLigneId="1" Numero="123"
            Description="Large Widget" Quantite="5"
            Price="22.00" />
          <CommandeLigne CommandeLigneId="2" Numero="234"
            Description="Medium Widget" Quantite="2"
            Price="12.50" />
        </CommandeLignes>
      </Commande>
      <Commande CommandeId="2" LivraisonDate="06-25-2006">
        <CommandeLignes>
          <CommandeLigne CommandeLigneId="5" Numero="432"
            Description="Small Widget" Quantite="30"
            Price="8.99" />
          <CommandeLigne CommandeLigneId="4" Numero="234"
            Description="Medium Widget" Quantite="2"
            Price="12.50" />
        </CommandeLignes>
      </Commande>
    </Commandes>
    <Factures>
      <Facture FactureId="6" Montant="99.37" />
      <Facture FactureId="7" Montant="147.50" />
    </Factures>
  </Client>

```

```
<Client ClientId="2" Nom="Tailspin Toys">
  <Commandes>
    <Commande CommandeId="8" LivraisonDate="07-11-2006">
      <CommandeLignes>
        <CommandeLigne CommandeLigneId="9" Numero="987"
          Description="Combo Widget" Quantite="2"
          Price="87.25" />
        <CommandeLigne CommandeLigneId="10" Numero="654"
          Description="Ugly Widget" Quantite="1"
          Price="2.00" />
      </CommandeLignes>
    </Commande>
    <Commande CommandeId="11" LivraisonDate="08-21-2006">
      <CommandeLignes>
        <CommandeLigne CommandeLigneId="12" Numero="999"
          Description="Pretty Widget" Quantite="50"
          Price="78.99" />
        <CommandeLigne CommandeLigneId="14" Numero="575"
          Description="Tiny Widget" Quantite="100"
          Price="1.20" />
      </CommandeLignes>
    </Commande>
  </Commandes>
  <Factures>
    <Facture FactureId="26" Montant="46.58" />
    <Facture FactureId="27" Montant="279.15" />
  </Factures>
</Client>
</Clients>
```

TreeView

```
<asp:XmlDataSource runat="server" ID="xdsClients" DataFile="~/App_Data/clients.xml" />

    <br />
    <asp:TreeView ID="TreeView2" runat="server"
DataSourceID="xdsClients"
ShowLines="True" ExpandDepth="0">
<DataBindings>
<asp:TreeNodeBinding DataMember="Client"
TextField="Nom" ValueField="ClientId" />
<asp:TreeNodeBinding DataMember="Order"
TextField="LivraisonDate" ValueField="CommandeId" />
<asp:TreeNodeBinding DataMember="CommandeLigne"
TextField="Description" ValueField="CommandeLigneId" />
<asp:TreeNodeBinding DataMember="Facture"
TextField="Montant" ValueField="FactureId"
FormatString="{0:C}" />
</DataBindings>
</asp:TreeView></div>
</form>
---
```

```
protected void TreeView2_SelectedNodeChanged(object sender, EventArgs e)
{
    Response.Write("Valeur:" + TreeView2.SelectedNode.Value);
}
```

Menu

Menu.Xml

```
<?xml version="1.0" encoding="utf-8" ?>
<MenuItems>
  <Home display="Home" url="/" />
  <Products display="Products" url="/products/">
    <SmallWidgets display="Small Widgets"
      url="/products/smallwidgets.aspx" />
    <MediumWidgets display="Medium Widgets"
      url="/products/mediumwidgets.aspx" />
    <BigWidgets display="Big Widgets"
      url="/products/bigwidgets.aspx" />
  </Products>
  <Support display="Support" url="/Support/">
    <Downloads display="Downloads"
      url="/support/downloads.aspx" />
    <FAQs display="FAQs"
      url="/support/faqs.aspx" />
  </Support>
  <AboutUs display="About Us" url="/aboutus/">
    <Company display="Company"
      url="/aboutus/company.aspx" />
    <Locations display="Location"
      url="/aboutus/locations.aspx" /></AboutUs></MenuItems>
```

Menu

```
<asp:XmlDataSource ID="XmlDataSource1" runat="server" DataFile="~/menu.xml"
    XPath="MenuItems/*">
</asp:XmlDataSource>
<br />
<asp:Menu ID="Menu2" runat="server" DataSourceID="XmlDataSource1"
    onmenuitemclick="Menu2_MenuItemClick">
</asp:Menu>

<asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
```

Code

```
protected void Menu2_MenuItemClick(object sender, MenuEventArgs e)
{
    Label1.Text = e.Item.ValuePath + ":" + e.Item.Value + ":" + e.Item.Text;
}
```

Exercice 1

- Le film sélectionné doit être affiché dans la zone de texte.

Independence Day ▼ Sélectionner

Independence Day

Exercice 2

- Utiliser un DataGridView

Aventure ▼ Sélectionner

titre	réalisateur
Star Wars	George Lucas
Jurassic Park	Steven Spielberg
Independence Day	Roland Emmerich

Ex3

- La première liste affiche la liste des films, la deuxième est utilisée pour sélectionner des films



Envoyer

Titanic
Jaws

Ex4

- Créer une page contenant :
 - Un GridView (affiche la liste des clients)
 - Un Détails: Affiche le client sélectionné dans le GridView et supporte les opérations Modifier et Nouveau.
 - Le GridView doit être actualisé après chaque modification à partir du contrôle DetailsView.
 - Pour le contrôle SqlDataSource Lié au DetailsView , définir un EmptyTemplate avec le texte « Aucun client sélectionné ». Et une commande de type New

ObjectDataSource

- Classe créée dans le dossier App_Code

```
public class Fournisseur
{private string cs =
ConfigurationManager.ConnectionStrings["csBase"].ToString();
public DataTable getFournisseurs()
{
    SqlDataAdapter sda=new SqlDataAdapter("select * from Fournisseurs",cs);
    DataSet ds = new DataSet("Base");
    sda.Fill(ds,"Fournisseurs");
    return ds.Tables["Fournisseurs"];}}
```

- Dans la page web, la propriété SelectMethod doit retourner un type qui implémente l'une des interfaces suivantes: IEnumerable, IListSource, IDataSource, ou IHierarchicalDataSource

```
<asp:ObjectDataSource ID="ObjectDataSource1" runat="server"
    SelectMethod="getFournisseurs"
    TypeName="Fournisseur"></asp:ObjectDataSource>
```

Paramètres

Plusieurs types de paramètres existent pour un ObjectDataSource: paramètres de type Select, Insert, Update, Delete et Filter. Ces paramètres fonctionnent avec les méthodes qui portent les même noms.

La source qui définit la valeur d'un paramètre peut provenir d'un objet Cookie, Control, Session, QueryString, Form, RouteData, ou Profile. La valeur d'un paramètre peut être définie aussi dans le code.

```
public class Clients
{ private string cs =
ConfigurationManager.ConnectionStrings["csBase"].ToString();
public DataTable getClients(string ville){ SqlDataAdapter sda = new
SqlDataAdapter("select * from Customers where City='" + ville + "'", cs);
    DataSet ds = new DataSet("C");
    sda.Fill(ds, "Clients");
    return ds.Tables["Clients"];}}
```

```
<asp:ObjectDataSource ID="ObjectDataSource1" runat="server"
    SelectMethod="getClients" TypeName="Clients">
    <SelectParameters>
        <asp:QueryStringParameter DefaultValue="Berlin" Name="ville"
            QueryStringField="ville" Type="String" />
    </SelectParameters>
</asp:ObjectDataSource>
```

- La classe Fournisseur

```
public class Fournisseur
{
    private string cs =
    ConfigurationManager.ConnectionStrings["cs
    Base"].ToString();
    public DataTable getFournisseurs()
    {
        SqlDataAdapter sda=new
        SqlDataAdapter("select ShipperId as id,
        CompanyName as nom, Phone as tel from
        Shippers",cs);
        DataSet ds = new DataSet("Base");
        sda.Fill(ds,"Fournisseurs");
        return ds.Tables["Fournisseurs"];}
    public void ajouter( string nom,
    string tel)
    {SqlConnection cnx=new
    SqlConnection(cs);
    cnx.Open();
        string sql="insert into shippers
    values ('" +nom + "', '" + tel + "')";
        SqlCommand cmd = new
    SqlCommand(sql,cnx);
        cmd.ExecuteNonQuery();
        cnx.Close();
    }
    public void modifier(int id, string
    nom, string tel)
    {
        SqlConnection cnx = new
    SqlConnection(cs);
        cnx.Open();
        string sql = "UPDATE shippers set
    CompanyName='" + nom + "',phone='" + tel
    + "' where ShipperId=" +id;
        SqlCommand cmd = new
    SqlCommand(sql,cnx);
        cmd.ExecuteNonQuery();
        cnx.Close();
    }
    public void supprimer(int id)
    {
        SqlConnection cnx = new
    SqlConnection(cs);
        cnx.Open();
        string sql = "Delete from shippers
    where ShipperId=" + id;
        SqlCommand cmd = new
    SqlCommand(sql,cnx);
        cmd.ExecuteNonQuery();
        cnx.Close();
    }
}}
```

• ObjectDataSource et DetailsView

```
<asp:DetailsView ID="DetailsView1" runat="server"
    DataSourceID="ObjectDataSource1 GridLines="None" >
  <Fields>
    <asp:CommandField ShowDeleteButton="True" ShowEditButton="True"
      ShowInsertButton="True" />
    <asp:BoundField DataField="id" HeaderText="ID" InsertVisible="False"
  />
    <asp:BoundField DataField="nom" HeaderText="Nom" />
    <asp:BoundField DataField="tel" HeaderText="Téléphone" />
  </Fields>
</asp:DetailsView>
<asp:ObjectDataSource ID="ObjectDataSource1" runat="server"
  SelectMethod="getFournisseurs" TypeName="Fournisseur"
  DeleteMethod="supprimer" InsertMethod="ajouter" UpdateMethod="modifier">
  <DeleteParameters>
    <asp:Parameter Name="id" Type="Int32" />
  </DeleteParameters>
  <InsertParameters>
    <asp:Parameter Name="nom" Type="String" />
    <asp:Parameter Name="tel" Type="String" />
  </InsertParameters>
  <UpdateParameters>
    <asp:Parameter Name="id" Type="Int32" />
    <asp:Parameter Name="nom" Type="String" />
    <asp:Parameter Name="tel" Type="String" />
  </UpdateParameters>
</asp:ObjectDataSource>
```

Les filtres

L'attribut FilterExpression peut contenir une expression de filtre valide qui est appliquée sur les données après leur extraction de la base de données.

```
<asp:ObjectDataSource ID="ObjectDataSource2" runat="server"«  
  TypeName="Clients" SelectMethod="getClients"FilterExpression="City='{0}'">  
<FilterParameters>  
<asp:QueryStringParameter Name="City" QueryStringField="City" Type="String"  
/></FilterParameters>  
</asp:ObjectDataSource>
```

Tri et pagination

```
public static DataTable getClients(string triCol, int debut, int nbEnr)  
{  
    if (nbEnr <= 10) nbEnr = 10;  
    if (triCol == "") triCol = "CompanyName";  
    string sql = "select * from Customers order by " + triCol ;  
    SqlDataAdapter sda = new SqlDataAdapter(sql, cs);  
    DataSet ds = new DataSet("C");  
    sda.Fill(ds, debut, nbEnr, "Clients");  
    return ds.Tables["Clients"];}  
---  
<asp:ObjectDataSource runat="server" ID="ods" TypeName="clients"  
  SelectMethod="getClients" SortParameterName="triCol"  
    EnablePaging="true" StartRowIndexParameterName="debut"  
  MaximumRowsParameterName="nbEnr"></asp:ObjectDataSource>  
  <asp:GridView ID="GridView1" runat="server"« DataSourceID="ods"« AllowPaging="True"«  
  PageSize="4" AllowSorting="true"> </asp:GridView>
```

SqlDataSource

Le contrôle SqlDataSource permet de configurer l'accès aux sources de données relationnelles: Sql Server, Oracle, OleDb, ODBC.

Propriétés

- Id,runat
- ConnectionString
- SelectCommand, SelectCommandType (Text, StoredProcedure)
- DataSourceMode: DataSet ou DataReader (ce dernier mode garde la connexion à la base de donnée ouverte).

```
<asp:SqlDataSource ID="sds" runat="server" ConnectionString="<%=  
ConnectionStrings:csBase %>" SelectCommandType="Text"  
SelectCommand="Select * from produits" DataSourceMode="DataReader">  
</asp:SqlDataSource>  
<asp:GridView ID="grd1" runat="server" DataSourceId="sds">  
</asp:GridView>
```

```
public partial class sqldatasource : System.Web.UI.Page  
{protected void Page_Init(object o, EventArgs e){  
    SqlDataSource sds1 =new SqlDataSource();  
    sds1.ID = "sds1";  
    sds1.ConnectionString=  
ConfigurationManager.ConnectionStrings["csBase"].ToString();  
    sds1.SelectCommand="Select * from clients";  
    this.Controls.Add(sds1);}  
protected void Page_Load(object sender, EventArgs e)  
{grd1.DataSourceID = "sds1";}}
```

Paramètres

Le contrôle SqlDataSource peut être configuré pour utiliser des paramètres dans les commandes Select, Update, Insert et Delete.

```
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%$ ConnectionStrings:csBase %>"
    SelectCommand="SELECT * FROM [Produits] WHERE ([Quantité] >
@Quantité)">
    <SelectParameters>
        <asp:QueryStringParameter DefaultValue="0" Name="Quantité"
            QueryStringField="quantite" Type="Int32" />
    </SelectParameters>
</asp:SqlDataSource>
```

Filtres et cache

Pour appliquer un filtre ou un cache, les données doivent être de type DataSet (DataSourceMode =DataSet), car le filtre est appliqué sur la propriété ADO.NET DataColumn ou DataView.RowFilter

XmlDataSource

- Un XmlDataSource peut être lié à un fichier Xml par sa propriété DataFile, ou bien dans le code la propriété Data peut contenir une chaîne xml.

```
<asp:XmlDataSource ID="XmlDataSource1" runat="server"  
    DataFile="~/App_Data/produits.xml"></asp:XmlDataSource>
```

Transformation d'un document XML

L'attribut TransformFile permet d'associer une feuille de style xsl

```
<asp:XmlDataSource ID="XmlDataSource1" runat="server"  
    TransformFile="~/App_Data/produits.xsl"  
    DataFile="~/App_Data/produits.xml" ></asp:XmlDataSource>
```

Filtres

L'attribut XPath permet de définir un filtre à l'aide d'une expression XPath.

```
<asp:XmlDataSource ID="XmlDataSource1" runat="server"  
    DataFile="~/App_Data/produits.xml"  
    TransformFile="~/App_Data/produits.xsl"  
    XPath="/Produits/Produit[Categorie='Categorie: boissons']" >  
</asp:XmlDataSource>
```

Remarque

Contrairement aux autres contrôles d'accès aux données, XmlDataSource ne possède pas des propriétés pour les opérations Insert, Update et Delete

SiteMapDataSource

- Le contrôle SiteMapDataSource permet la connexion aux données de navigation définies dans le plan de site (web.sitemap) . Il est utilisé par les contrôles Menu et TreeView

Web.sitemap

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
  <siteMapNode url="" title="Accueil" description="">
    <siteMapNode url="produits.aspx" title="Produits" description="">
      <siteMapNode url="detailsProduit.aspx"
        title="Détails produit" description="" />
    </siteMapNode>
    <siteMapNode url="services.aspx" title="Services" description="" />
    <siteMapNode url="apropos.aspx" title="A propos" description="" />
  </siteMapNode>
</siteMap>
<asp:SiteMapDataSource
ID="SiteMapDataSource1"
runat="server" />
<asp:Menu ID="Menu1"
runat="server"
DataSourceID="SiteMapDataSource1">
</asp:Menu>
```

Propriétés

- **StartingNodeUrl**: définit le nœud du plan de site qui sera la racine des données du contrôle `SiteMapDataSource`
- **ShowStartingNode**
- **StartFromCurrentNode**: la racine sera définie à partir de la page courante dans le navigateur.
- **StartingNodeOffset**: permet de déplacer le nœud de démarrage par rapport à la position courante.