Px Framework v.1.07 for ASP.NET

Komponenty pre rýchlu prácu z databázou

OBSAH

1.Základné komponenty Px Frameworku (Px Controls), stručný popis programovania a práce s nimi

1.1. **PxWebQuery –** komponenta pre prácu s databázou, základný popis
1.2. **PxWebQuery** - zadávanie prepojení (relácii) komponenty PxWebQuery s ostatnými tabuľkami
1.3. **PxSuperGrid –** komponenta pre priame zobrazenie dát v tabuľke
1.4. **PxEdit** – komponenta, ktorá slúži na editáciu dát, obdoba komponenty TextBox
1.5. **PxComboBox** – komponenta, ktorá slúži na výber dát zo zoznamu, obdoba komponenty DropDownList
1.6. **PxCheckBox** – komponenta, ktorá slúži na zaškrtnutie (select / unselect) hodnoty
1.7. **PxFlyComboBox** – séria comboboxov, zapojených za sebou, vhodný pre prácu zo štruktúrovanými dátami (napr. výber kategórii a podkategórii)
1.8. **PxGreatRepeater** – komponenta pre zadávanie dát, ktoré sa svojou štruktúrou opakujú, maximálny počet hodnôt je limitovaný
1.9. **PxDbNavigator –** komponenta pre prácu z komponentov PxWebQuery, posúvanie kurzora riadku a pod.
1.10. **PxJSDatePicker** – komponenta pre zadanie dátumu, postavená na JavaScripte
1.11. **PxDatePicker** – komponenta pre zadanie dátumu
1.12. **PxMemo** – komponenta, ktorá slúži na priamu editáciu dát, rozšírenie komponenty PxEdit
1.13. **PxLabel** – komponenta pre zobrazenie dát

2. Inštalácia Px Frameworku

2.1. Inštalácia Px Frameworku na klientsky počítač
2.2. Inštalácia Px Frameworku na server

3. Detailný popis programovania s komponentami Px Frameworku

3.1. PxWebQuery – komponenta pre prácu s databázou Oracle, detailný popis

3.1.1. Načítanie dát z databázy Oracle, pomocou komponenty PxWebQuery
3.1.2. Programové vkladanie nového riadku do databázy pomocou komponenty PxWebQuery
3.1.3. Programová editácia existujúceho riadku v databáze pomocou komponenty PxWebQuery
3.1.4. Programové zmazanie existujúceho riadku v databáze pomocou komponenty PxWebQuery
3.1.5. Načítanie hodnôt z komponenty PxWebQuery cez cyklus while
3.1.6. Vyhľadávanie riadku v komponente PxWebQuery, podľa zadanej hodnoty a názvu stĺpca v ktorom sa má vyhľadávať
3.1.7. Výber, selekcia viacerých riadkov, ktoré vyhovujú výberovým kritériám
3.1.8. Udalosti (Eventy) komponenty PxWebQuery
3.1.9. Vytvorenie nového dynamického stĺpca(field) v tabuľke komponenty PxWebQuery
3.1.10. Procedúra ReOpen komponenty PxWebQuery a znovu načítanie dát do komponenty PxWebQuery

3.1.11. Validácia, kontrola zadaných hodnôt prostredníctvom komponenty PxWebQuery a ostatných vizuálnych komponent(PxEdit, PxComboBox a atď.)
3.1.12. Zistenie verzie Px Frameworku
3.1.13. Nastavenie jazykovej mutácie Px Frameworku

3.2. Práca zo štýlmi, kaskádové štýly(CSS)

3.2.1. PxEdit – práca zo štýlmi, kaskádové štýly(CSS)
3.2.2. PxJSDatePicker – práca zo štýlmi, kaskádové štýly(CSS)
3.2.3. PxGreatRepeater – práca zo štýlmi, kaskádové štýly(CSS)

4.1 Špecifikácia nasadenia Px Frameworku pod rôznymi databázovými platformami (Oracle, MS SQL, MySQL, FireBird, InterBase)

4.1.1. Špecifikácia nasadenia Px Frameworku pod databázov Oracle
4.1.2. Špecifikácia nasadenia Px Frameworku pod databázov MS SQL
4.1.3. Špecifikácia nasadenia Px Frameworku pod databázov Mysql
4.1.4. Špecifikácia nasadenia Px Frameworku pod databázov FireBird
4.1.5. Špecifikácia nasadenia Px Frameworku pod databázov InterBase
4.1.6. Špecifikácia nasadenia Px Frameworku všeobecne

5.1 **Pomocné komponenty Px Frameworku**
5.1.1. Komponenta pre načítanie dát z databázy - SQLCoreQuery

1.1 PxWebQuery – komponenta pre prácu s databázou

Komponenta *PxWebQuery* je komponentou, ktorá umožňuje rýchlu prácu z databázou, natiahnutie dát z databázy, ich zobrazenie a úpravu(vkladanie riadku, editovanie a mazanie), prípadné hľadanie a výber dát. Súčasťou komponenty *PxWebQuery* je objekt *DataTable*, do ktorého sa naťahujú dáta, definované SQL príkazom.

V súbore *.aspx sa komponenta *PxWebQuery* definuje nasledovne:

```
<Prx:PxWebQuery ID="wquAdresar" runat="server" Value="wquAdresar"/>
```

Ak chceme, aby komponenty Px Frameworku boli viditeľné v aspx súboroch, musíme si v hlavičke súboru zadefinovať „Prx“ prefix. Toto vykonáme nasledovne:

```
<%@ Register TagPrefix="Prx" Namespace="PxControls" Assembly="PxControls" %>
```

Keď máme toto vykonané, môžeme v súbore *.aspx.cs s komponentou *PxWebQuery* pracovať nasledovne. Najprv si do komponenty zadáme ConnectionString a potom SQL príkaz:

```
wquAdresar.ConnectString = "User Id=adr;Password=aa;Data Source=xe;";
wquAdresar.SQLSelect = "select idadresar, name, address, create_date from Adresar";
```

Potom cez príkaz „Open“, načítame do komponenty dáta z databázy, a môžeme ju pripojiť na komponentu *PxSuperGrid*, v ktorej sa zobrazia načítané dáta.

Príklad č.1.

Súbor adresar.aspx.cs:

```
protected void Page_Load(object sender, EventArgs e)
{
 if ((!IsPostBack)&&(wquAdresar.Active==false))
 {
   wquAdresar.ConnectString = "User Id=adr;Password=aa;Data Source=xe;";
   wquAdresar.SQLSelect = "select idadresar , name, address, create_date from Adresar";

   wquAdresar.Open();
 }
 grdAdresar.PxDataSource = wquAdresar;
 grdAdresar.DataBind();
 }
```

Súbor adresar.aspx:

```
<Prx:PxWebQuery ID="wquAdresar" runat="server" Value="wquAdresar"/>

<Prx:PxSuperGrid ID="grdAdresar" runat="server"
    PageSize="4"
    PxEditFormName=""
```

```
            PxInfoFormName=""
            PxVisibleButtons="SIDEO">
        </Prx:PxSuperGrid>
```

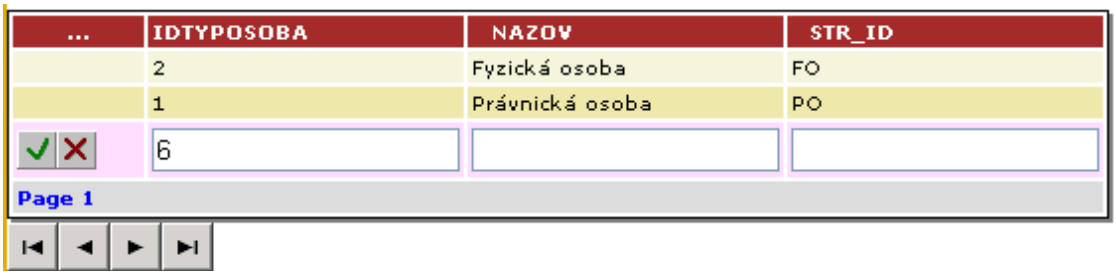
Komponenta *PxWebQuery* automaticky poskytuje pri prepojený s *PxSuperGridom* funkcie, ako vkladanie dát, editovanie, mazanie, sortovanie a stránkovanie(paging).
Na to aby ste prostredníctvom *PxWebQuery* vkladali nove riadky, potrebujete vytvoriť sequenciu, z ktorej si komponenta *PxWebQuery* načíta jedinečné ID pre svoj primárny kľúč. Názov sequencie, by ste mali vytvoriť nasledujúcim spôsobom.

SEQ_“názov stĺpca primárneho kľúča“

Preto, keď tvoríte nové tabuľky nedávajte názov stĺpcu s primárnym kľúčom len „ID“, ale napríklad ID+“názov tabuľky“.
V našom príklade, aby nám fungovalo aj vkladanie dát do databázy, vytvoríme sequenciu SQL príkazom:

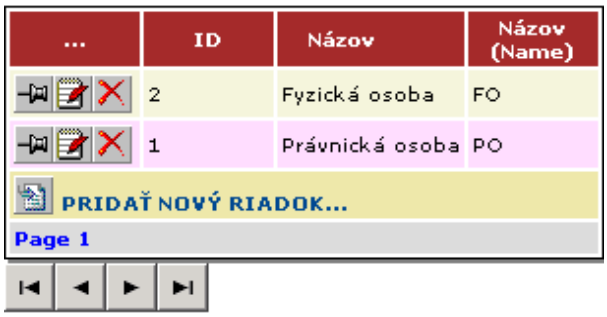
```
CREATE SEQUENCE SEQ_IDADRESAR
  START WITH 1
  INCREMENT BY 1
  MAXVALUE 1E18
  NOMINVALUE
  NOORDER
  NOCACHE
  NOCYCLE;
```

Pozor !!! Samozrejme pod inými databázovými platformami, sa sequencia vytvára inak, viac sa môžete o tomto dočítať v štvrtej časti „ Špecifikácia nasadenia Px Frameworku pod rôznymi databázovými platformami (Oracle, MS SQL, MySQL, FireBird, InterBase)**”.**



...	IDTYPOSOBA	NAZOV	STR_ID
	2	Fyzická osoba	FO
	1	Právnická osoba	PO
	6		

Page 1



...	ID	Názov	Názov (Name)
	2	Fyzická osoba	FO
	1	Právnická osoba	PO

PRIDAŤ NOVÝ RIADOK...

Page 1

1.2. PxWebQuery - zadávanie prepojení (relácii) komponenty PxWebQuery s ostatnými tabuľkami

1.2.1. AddParamKey – pripojenie jednoduchého číselníka k hlavnej tabuľke

AddParamKey slúži na prepojenie tabuľky z číselníkom. Metóda *AddParamKey* je metódou komponenty *PxWebQuery*. Filozofia Px Frameworku je taká, aby čo najviac informácii bolo definovaných na komponente *PxWebQuery*. Týmto sa vlastne šetrí programovanie a zápis pri komponentách *PxSuperGrid*, *PxEdit*, *PxComboBox* a pod. Všetky tieto komponenty si požičiavajú dáta a nastavenie z komponenty *PxWebQuery*. Preto na komponente *PxWebQuery* máme množstvo procedúr a metód, prostredníctvom ktorých definujeme rôzne zobrazenie a prepojenie medzi tabuľkami.

Taká najjednoduchšia metóda prostredníctvom, ktorej sa definuje prepojenie medzi hlavnou tabuľkou a číselníkom je metóda *AddParamKey*.

Tuto definíciu môžeme zapísať do súboru Adresar.aspx.cs.
Príklad č.1.

```
protected void Page_Load(object sender, EventArgs e)
{
  if ((!IsPostBack)&&(wquAdresar.Active==false))
  {
    string sSQLText1 = "select IDPravForm as Key, Name as Value from PravForm Order By Value";

    wquAdresar.ConnectString = "User Id=adr;Password=aa;Data Source=xe;";
    wquAdresar.SQLSelect = "select idadresar , name, address, idpravnaforma, create_date from Adresar";

    wquAdresar.AddParamKey("idpravnaforma", "Key", "Value", "Value", "Key", sSQLText1);

    wquAdresar.Open();

    wquAdresar.Columns["idadresar"].Caption = "ID";
    wquAdresar.Columns["name"].Caption = "Meno Osoby";
    wquAdresar.Columns["idpravnaforma"].Caption = "Právna Forma";
    wquAdresar.Columns["address"].Caption = "Adresa";
    wquAdresar.Columns["create_date"].Caption = "Dátum založenia";
  }
  else
  {
  }
    grdAdresar.PxDataSource = wquAdresar;
    grdAdresar.DataBind();
}
```

Popis Parametrov metódy *AddParamKey*:

C# syntax:

```
public void AddParamKey(string aFieldName, string aFieldNameKey, string aFieldNameValue,
        string aFieldToView, string aFieldToDB,
```

```
string aSQLText);
```

Popis parametrov:

aFieldName – Meno stĺpca v tabuľke na ktorý sa aplikuje metóda AddParamKey
aFieldNameKey – Meno stĺpca číselníka, ktorý reprezentuje kľúč číselníka (toto meno stĺpca sa dá zistiť z Sql príkazu, ktorý sa zadáva do parametra **aSQLText**)
aFieldNameValue – Meno stĺpca číselníka, ktorý reprezentuje hodnotu(Value) číselníka (toto meno stĺpca sa dá zistiť z Sql príkazu, ktorý sa zadáva do parametra **aSQLText**)
aFieldToView – tu sa zadáva meno stĺpca, ktorého hodnota sa bude zobrazovať v Gride(*PxSuperGrid*)
aFieldToDB - tu sa zadáva meno stĺpca, ktorého hodnota bude zapísaná do databázy
aSQLText – tu sa zadefinuje SQL príkaz pre načítanie číselníka, jednotlivé stĺpce je vhodne označiť ako Key a Value, kvôli lepšej prehľadnosti

Ako vyzerá komponenta PxSuperGrid pred aplikáciou metódy *AddParamKey*:

...	ID	Meno Osoby	Adresa	Právna Forma	Dátum založenia
	1	Peter ŠKODA	Banská Bystrica	1	10. 10. 2009 0:00:00
	2	Ján Kolenička	Nitra	1	11. 2. 2010 0:00:00
	3	Palma Duffy	Bratislava	2	4. 5. 2009 0:00:00
	4	Jan Shrek	Dolne Kosiby	1	11. 8. 2010 0:00:00

PRIDAŤ NOVÝ RIADOK...
Page 1 [2]

Ako vyzerá komponenta PxSuperGrid po aplikácii metódy *AddParamKey*:

...	ID	Meno Osoby	Adresa	Právna Forma	Dátum založenia
	1	Peter ŠKODA	Banská Bystrica	Fyzická osoba	10. 10. 2009 0:00:00
	2	Ján Kolenička	Nitra	Fyzická osoba	11. 2. 2010 0:00:00
	3	Palma Duffy	Bratislava	Právnická osoba	4. 5. 2009 0:00:00
	4	Jan Shrek	Dolne Kosiby	Fyzická osoba	11. 8. 2010 0:00:00

PRIDAŤ NOVÝ RIADOK...
Page 1 [2]

Po aplikácii metódy *AddParamKey*, na stĺpec idpravnaforma, bol kľúč v tejto tabuľke nahradený hodnotou value, z číselníka definovaného SQL príkazom v parametre *aSQLText* v metóde *AddParamKey*. Pozrite si podrobne v príklade ako bol zostavený SQL príkaz, cez ktorý sa potom, nahradzovala kľúčová celočíselná hodnota, hodnotou textu z číselníka.

Ďalšia funkcionalita komponenty PxSuperGrid pri zadaný parametra *AddParamKey,* umožňuje pri editácii v Gride nahradiť automaticky komponentu *TextBox* komponentou *DropDownList*. Viac uvidíte na obrázku dole, že na stĺpec na ktorý sa uvalil parameter

AddParamKey, je nahradený *ComboBoxom*, aj s predom načítanými hodnotami, ktoré si môžeme zvoliť, a ktoré sú pri potvrdení aj automaticky uložené do databázy.

Ďalšie parametre, ako *AddParamWebQuery*, *AddParamFlyComboBox*, *AddParamGreatRepeater* a pod., budú popísané v druhej časti tohto manuálu.

1.2.2. AddParamWebQuery – pripojenie jednoduchého číselníka k hlavnej tabuľke

Metóda AddParamWebQuery je podobná, ako metóda AddParamKey, pričom ako zdroj dát je využívaná komponenta PxWebQuery. Použitie metódy AddParamWebQuery, je vhodné vtedy, keď číselník obsahuje dáta ktoré sa dynamicky menia. Pokiaľ v číselníku máme zadané údaje, ktoré sa dlhodobo nemenia, stačí použiť na prepojenie hlavnej tabuľky s číselníkom metódu AddParamKey.

Popis Parametrov metódy *AddParamWebQuery*:

C# syntax:

```
public void AddParamWebQuery(string aFieldName, string aFieldNameKey, string aFieldNameValue,
        string aFieldToView, string aFieldToDB,
        string aPxWebQueryName)
```

Popis parametrov:

aFieldName – Meno stĺpca v tabuľke na ktorý sa aplikuje metóda AddParamWebQuery
aFieldNameKey – Meno stĺpca číselníka, ktorý reprezentuje kľuč číselníka (toto meno stĺpca sa dá zistiť z Sql príkazu, ktorý sa zadáva do komponenty PxWebQuery v ktorej je načítaný číselník)
aFieldNameValue – Meno stĺpca číselníka, ktorý reprezentuje hodnotu(Value) číselníka (toto meno stĺpca sa dá zistiť z Sql príkazu, ktorý sa zadáva do komponenty PxWebQuery v ktorej je načítaný číselník)
aFieldToView – tu sa zadáva meno stĺpca, ktorého hodnota sa bude zobrazovať v Gride(*PxSuperGrid*)
aFieldToDB - tu sa zadáva meno stĺpca, ktorého hodnota bude zapísaná do databázy
aPxWebQueryName – Názov komponenty PxWebQuery v ktorej je cez SQL príkaz načítaný číselník, jednotlivé stĺpce je vhodne označiť ako Key a Value, kvôli lepšej prehľadnosti

```
wquAdresar.AddParamWebQuery("idpravnaforma", "Key", "Value", "Value", "Key", "wquPravForm");
```

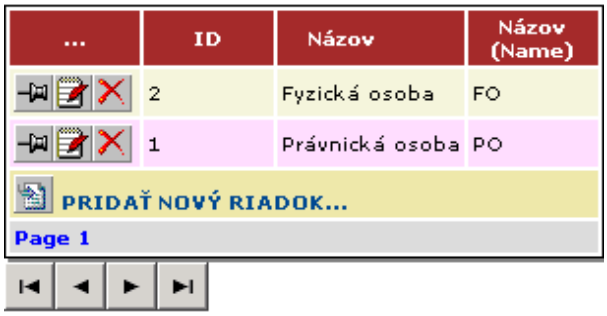
1.3. PxSuperGrid – komponenta pre priame zobrazenie dát v tabuľke

Komponenta PxSuperGrid slúži na priame zobrazenie dát v tabuľke, prostredníctvom komponenty PxWebQuery. Pomocou komponenty PxSuperGrid môžete pracovať z dátami nasledovným spôsobom: Môžete vkladať nové riadky, editovať alebo mazať už existujúce riadky. Ďalej Vám komponenta umožňuje automatické sortovanie a stránkovanie(paging).

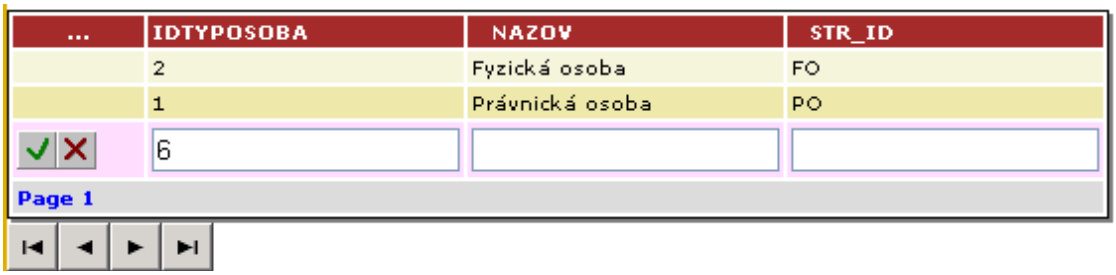
Popis funkcionality jednotlivých tlačítok PxSuperGridu:

– toto tlačítko slúži na editáciu, už existujúceho riadku
– toto tlačítko slúži na mazanie existujúceho riadku
– toto tlačítko slúži na selekciu, výber riadku
– toto tlačítko slúži pre zavolanie informačného formulára, prevažne sa využíva na zobrazenie detailu daného riadku

PRIDAŤ NOVÝ RIADOK... - toto tlačítko slúži na pridávanie nového riadku



Takto vyzerá editácia prostredníctvom komponenty PxSuperGrid.



Nasledujúci príklad Vám ukáže, ako sa pracuje s PxSuperGridom. Najprv si dáta z databázy natiahnete do komponenty PxWebQuery, ktorú otvoríte príkazom Open() a potom túto komponentu nabindujete, pripojíte na komponentu PxSuperGrid.

Príklad č.2.
Súbor adresar.aspx.cs:

```
protected void Page_Load(object sender, EventArgs e)
{
```

```
if ((!IsPostBack)&&(wquAdresar.Active==false))
{
    wquAdresar.ConnectString = "User Id=adr;Password=aa;Data Source=xe;";
    wquAdresar.SQLSelect = "select idadresar , name, address, create_date from
    Adresar";

    wquAdresar.Open();

}
    grdAdresar.PxDataSource = wquAdresar;
    grdAdresar.DataBind();
}
```

Súbor adresar.aspx:

```
<Prx:PxWebQuery ID="wquAdresar" runat="server" Value="wquAdresar"/>

<Prx:PxSuperGrid ID="grdAdresar" runat="server"
  PageSize="4"
  PxVisibleButtons="SIDEO">
</Prx:PxSuperGrid>
```

Ak chcete zmeniť popisok v tabuľke PxSuperGridu(Title), môžete to vykonať zápisom nasledujúceho kódu v súbore *.aspx.cs, priradeným do komponenty PxWebQuery, property Caption.

```
wquAdresar.Columns["NAME"].Caption = "Priezvisko  a Meno";
```

Ak chceme meniť zobrazenie tlačítok v *PxSuperGride*(ich zobrazenie alebo skrytie), použijeme property *PxVisibleButtons* komponenty *PxSuperGridu*.

```
grdAdresar.PxVisibleButtons="SIDEO";
```

Tu v ďalšom je popísaná legenda, podľa ktorej sa zobrazujú jednotlivé tlačítka, pokiaľ je tam zapísané nejaké písmeno, tak sa v PxSuperGride, zobrazí príslušné tlačítko

```
S – select, výber riadku
I – insert, zadanie, alebo vloženie nového riadku
E – editovanie už existujúceho riadku
D – mazanie  už existujúceho riadku
O - otvorenie informačného formulára
```

Pokiaľ nechceme prevádzať editáciu a vkladanie nového riadku priamo v komponente *PxSuperGrid*, stači keď, do property *PxEditFormName* zadáme cestu, alebo názov formulára kde bude táto editácia vykonaná.

```
grdAdresar.PxEditFormName="AddAdresar.aspx";
```

Túto property si môžeme zadefinovať aj v súbore Adresar.aspx v tele definície *PxSuperGridu*.

```
<Prx:PxSuperGrid ID="grdAdresar" runat="server"
        PageSize="4"
        PxEditFormName="AddAdresar.aspx"
        PxInfoFormName="InfoAdresar.aspx"
        PxVisibleButtons="SIDEO">
</Prx:PxSuperGrid>
```

Po stlačení tlačítka „editácia“ , alebo „pridať riadok“ v komponente *PxSuperGridu*, bude otvorení formulár AddAdresar.aspx. Formulár AddAdresar.aspx si môžeme vyskladať z editačných prvkov *PxEdit* a *PxComboBox*.
Viac, k tejto téme bude popísane pri popise komponent *PxEdit* a *PxComboBox*.

Popis tlačítok PxSuperGridu:

- – toto tlačítko slúži na editáciu, už existujúceho riadku
- – toto tlačítko slúži na mazanie existujúceho riadku
- – toto tlačítko slúži na selekciu, výber riadku
- – toto tlačítko slúži pre zavolanie informačného formulára, prevažne sa využíva na zobrazenie detailu daného riadku

PRIDAŤ NOVÝ RIADOK... - toto tlačítko slúži na pridávanie nového riadku

Podobne si môžeme zadefinovať aj formulár kde sa, bude zobrazovať len informácia o danom riadku, kde sa bude zobrazovať detail daného riadku.

```
PxInfoFormName="InfoAdresar.aspx";
```

Formulár InfoAdresar.aspx, si pre zmenu môžeme vyskladať z komponent PxLabel

Ak chceme aby sa nám v komponente PxSuperGrid zobrazoval aj vyselektovaní, vybraný riadok, zabezpečme to zadaním nasledujúceho riadku do komponenty PxSuperGrid:

```
<SelectedItemStyle BackColor="#ffddff" />
```

Takže celý zápis definície komponenty PxSuperGrid v súbore Adresar.aspx bude vyzerať nasledovne:

```
<Prx:PxSuperGrid ID="grdAdresar" runat="server"
        PageSize="4"
        PxEditFormName="AddAdresar.aspx"
        PxInfoFormName="InfoAdresar.aspx"
        PxVisibleButtons="SIDEO">
  <SelectedItemStyle BackColor="#ffddff" />

</Prx:PxSuperGrid>
```

Na to, aby sme zredukovali počet zobrazených stĺpcov v Gride použijeme property VisibleFields komponenty PxSuperGrid, viac ukážka dole:

```
grdAdresar.VisibleFields = "idadresar;name;idpravnaforma;address;miesto;telefon";
```

Do property VisibleFields sme priradili názvy stĺpcov, oddelených bodkočiarkou, ktoré sa majú zobraziť pri zobrazený v komponente PxSuperGrid-u. Ak v property VisibleFields, nie je priradené žiadna hodnota, v komponente PxSuperGrid-u sa nám zobrazia všetky stĺpce(fields). Samozrejme všetky stĺpce sa nám v komponente PxSuperGrid zobrazia presne v tom poradí, v akom poradí sme ich zadali do property VisibleFields.
Prostredníctvom property VisibleFields, môžeme zadefinovať taktiež dĺžku stĺpca, tým, že za každým stĺpcom, urobíme dvojbotku, a za ňou zadáme číslo v pixeloch, ktoré má definovať dĺžku stĺpca. Viac príklad dole:

```
grdAdresar.VisibleFields = "idadresar:50;name;idpravnaforma;address;miesto:100;telefon:150";
```

Property VisibleFields, na to aby správne fungovala, musí byť definovaná pred pripojením a bindovaním na komponentu PxSuperGrid.

```
grdAdresar.VisibleFields = "idadresar;name;idpravnaforma;address;miesto;telefon";
grdAdresar.PxDataSource = wquAdresar;
grdAdresar.DataBind();
```

1.4. PxEdit – komponenta, ktorá slúži na editáciu dát, obdoba komponenty TextBox

Pomocou komponenty PxEdit môžeme automaticky editovať stĺpec riadka tabuľky, ktorý je natiahnutý v komponente PxWebQuery. Vytvoríme si súbor AddAdresar.aspx, a do neho zapíšeme nasledujúcu definíciu:

Súbor AddAdresar.aspx:

```
<Prx:PxWebQuery ID="wquAdresar" runat="server" Value="wquAdresar"/>
```

```
<asp:Label ID="lblAddEditPanel" Font-Size="18pt" Font-Bold="true" runat="server"
  Text="Label"></asp:Label>
 <br /><br />

<Prx:PxEdit ID="edtIDADRESAR" runat="server" AddTableRow="True" TableBegin="True" >
</Prx:PxEdit>
<Prx:PxEdit ID="edtName" runat="server" AddTableRow="True"  ></Prx:PxEdit>
<Prx:PxEdit ID="edtADDRESS" runat="server" AddTableRow="True"  ></Prx:PxEdit>
<Prx:PxComboBox ID="cmbPravnaForma" runat="server" AddTableRow="True"  ></Prx:PxComboBox>
<Prx:PxJSDatePicker ID="edtCREATE_DATE" runat="server" AddTableRow="True" TableEnd="True">
</Prx:PxJSDatePicker>
<br />
<asp:Button ID="btnOK" runat="server" Text="Ok" Width="60" onclick="ButtonOk_Click" />
<asp:Button ID="btnStorno" runat="server" Text="Storno" Width="60" onclick="ButtonStorno_Click" />
```

V súbore AddAdresar.aspx.cs, si zase zadefinujeme prepojenie jednotlivých komponent *PxEdit* s komponentou *PxWebQuery*, a jej prepojenie na konkrétny stĺpec tabuľky. Ďalej v tomto súbore sú zadefinované metódy tlačítok „Ok“ a „Storno“. V prvom prípade, pri tlačítku „Ok“, sa všetky zmeny v komponentách PxEdit potvrdia príkazom Post(), v druhom prípade pri tlačítku „Storno“, sa všetky zmeny anulujú príkazom Cancel().
Ak sú oba príkazy úspešne vykonané, potom príkazom GoBackPreviousPage() sme vrátení do formulára Adresar.aspx. Ak ukladanie dát, bolo neúspešné, ostaneme v pôvodnom formulári, kde za konkrétnou komponentou PxEdit, bude popísaný dôvod prečo nedošlo ku automatickému uloženiu. Komponenta PxEdit podporuje aj automatickú validáciu, viac o automatickej validácii a ako sa s ňou pracuje, nájdete v druhej časti tohto manuálu.

Súbor AddAdresar.aspx.cs:

```
protected void Page_Load(object sender, EventArgs e)
 {
    if (wquAdresar.StateQuery == PxWebQuery.eStateQuery.sqInsert)
       lblAddEditPanel.Text = "Pridať adresu";
    else lblAddEditPanel.Text = "Editovať adresu";

    edtIDADRESAR.PxDataSource = wquAdresar;
    edtIDADRESAR.FieldName = wquAdresar.Columns["IDADRESAR"].ToString();

    edtName.PxDataSource = wquAdresar;
    edtName.FieldName = wquAdresar.Columns["NAME"].ToString();

    edtADDRESS.PxDataSource = wquAdresar;
    edtADDRESS.FieldName = wquAdresar.Columns["ADDRESS"].ToString();

    cmbPravnaForma.PxDataSource = wquAdresar;
    cmbPravnaForma.FieldName = wquAdresar.Columns["idpravnaforma"].ToString();

    edtCREATE_DATE.PxDataSource = wquAdresar;
    edtCREATE_DATE.FieldName = wquAdresar.Columns["CREATE_DATE"].ToString();
 }

 protected void ButtonOk_Click(object sender, EventArgs e)
 {
    wquAdresar.Post();
    wquAdresar.GoBackPreviousPage();
 }

 protected void ButtonStorno_Click(object sender, EventArgs e)
 {
```

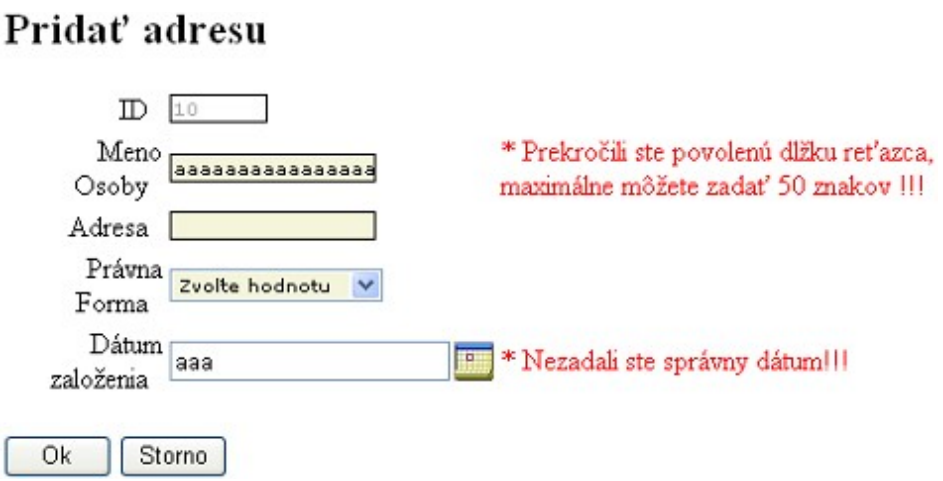
```
        wquAdresar.Cancel();
        wquAdresar.GoBackPreviousPage();
    }
```

Výsledný formulár pre editáciu riadka tabuľky vyzerá nasledovne:

Na to aby bol zavolaný formulár AddAdresar.aspx, musíme vo formuláry Adresar.aspx pre komponentu PxSuperGrid priradiť property PxEditFormName meno formulára v ktorom sa budu editovať dáta za daný zvolený riadok.

```
<Prx:PxSuperGrid ID="grdAdresar" runat="server"
    PageSize="4"
    PxEditFormName="AddAdresar.aspx"
    PxInfoFormName="InfoAdresar.aspx"
    PxVisibleButtons="SIDEO">
  <SelectedItemStyle BackColor="#ffddff" />

</Prx:PxSuperGrid>
```

Po stlačení tlačítka „editovať“ v komponente PxSuperGridu , sa nám otvorí formulár AddAdresar.aspx.
Tam môžeme zadávať nové hodnoty, poprípade meniť staré hodnoty, po stlačení tlačítka „Ok“, sú tieto hodnoty uložené.
Komponenta PxEdit(ale aj PxComboBox, PxFlyComboBox, PxJSDatePicker a pod.) obsahuje automatickú validáciu. Ak stĺpec tabuľky, ktorý je pripojený na komponentu PxEdit je typu **string,** je automaticky prevádzaná validácia dĺžky reťazca, ak je typu **int**, validuje sa či bolo zadané celé číslo, ak je typu **date**, validuje sa či bol zadaný validný dátum a pod. Viac o automatickej validácii sa dočítate v druhej časti knihy.
Na obrázku je príklad, kde sme zadali reťazec, ktorý svojou dĺžkou presahoval dĺžku reťazca stanoveného v databáze. Ďalej sme zadali dátum, ktorý nebol validný.



Polohovanie a zarovnanie prvkov PxEdit:

Keď do súboru AddAdresar.aspx, zadáme komponenty PxEdit, jednu za druhou, dôjde nám ku nesúmernému rozhádzaniu prvkov, viac uvidíte na obrázku:

Toto nesúmerné rozhádzanie môžeme napraviť nasledujúcim spôsobom:
Ku každému prvku zadefinujeme property AddTableRow a tu nastavíme na hodnotu True;

```
<Prx:PxEdit ID="edtIDADRESAR" runat="server" AddTableRow="True" ></Prx:PxEdit>
```

Každá komponenta PxEdit je zložená z nasledujúcich prvkov v tomto poradí:

Label + TextBox + Label alebo Button

Táto zapnutá property nám medzi tieto komponenty vloží tabuľkový riadok.
V praxi to vyzerá nasledovne:

<tr><td>Label + <td>TextBox +<td> Label alebo Button<td>

Pri prvom prvku nastavíme ešte property TableBegin na True. V praxi to vyzerá nasledovne:

```
<Prx:PxEdit ID="edtIDADRESAR" runat="server" AddTableRow="True" TableBegin="True" ></Prx:PxEdit>
```

Pri poslednej komponente PxEdit v poradí nastavíme zase property TableEnd na True. V praxi to vyzerá nasledovne:

```
<Prx:PxEdit ID="edtCREATE_DATE" runat="server" AddTableRow="True" TableEnd="True"  ></Prx:PxEdit>
```

Takto to môžeme aplikovať na celý kód, a ten bude vyzerať nasledovne:

```
<Prx:PxEdit ID="edtIDADRESAR" runat="server" AddTableRow="True" TableBegin="True" ></Prx:PxEdit>
<Prx:PxEdit ID="edtName" runat="server" AddTableRow="True"  ></Prx:PxEdit>
<Prx:PxEdit ID="edtADDRESS" runat="server" AddTableRow="True"  ></Prx:PxEdit>
<Prx:PxComboBox ID="cmbPravnaForma" runat="server" AddTableRow="True"  ></Prx:PxComboBox>
<Prx:PxJSDatePicker ID="edtCREATE_DATE" runat="server" AddTableRow="True" TableEnd="True"  >
</Prx:PxJSDatePicker>
```

Keď budeme mať property AddTableRow, TableBegin a TableEnd správne nastavené, potom, by sa nám mali prvky správne zoradiť a zarovnať, viacej na obrázku:

Názov popiskov Caption alebo názvy stĺpcov v gride(Title) môžete zadať centrálne, pri definícii komponenty PxWebQuery, kde hodnota v hranatých zátvorkách je názov stĺpca:

```
wquAdresar.Columns["NAME"].Caption = "Priezvisko  a Meno";
```

Zmena popisku komponenty PxEdit (Label):

Zmenu popisku komponenty PxEdit môžeme vykonať, okrem centrálnej definície, nasledujúcim spôsobom:

```
edtIDADRESAR.Caption = "Main text";
```

Zmena popisku At Post komponenty PxEdit (Label):

Komponenta PxEdit obsahuje ešte ďalší popisok, ktorý sa nachádza za komponentov TextBox. Ku tomuto popisku sa dá pristúpiť cez property *RCaption*.

```
edtIDADRESAR.RCaption = "(Primary Key tbl Adresar)";
```

Výsledný tvar zdrojového kódu bude potom nasledujúci:

```
edtIDADRESAR.PxDataSource = wquAdresar;
edtIDADRESAR.FieldName = wquAdresar.Columns["IDADRESAR"].ToString();
```

```
edtIDADRESAR.Caption = "Main text";
edtIDADRESAR.RCaption = "(Primary Key tbl Adresar)";
```

Výsledok si môžete pozrieť na obrázku:

1.5. PxComboBox – komponenta, ktorá slúži na výber dát zo zoznamu, obdoba komponenty DropDownList

Komponenta PxComboBox pracuje obdobne, alebo ma podobné property, ako komponenta PxEdit. Preto nemá zmysel sa nimi podrobne zaoberať, môžete si prečítať kapitolu predchádzajúcu o komponente PxEdit. Veci, ktoré boli podrobne popísané pri komponente PxEdit, budú tuná prejdené len telegraficky, zbežne.

Pomocou komponenty PxComboBox môžeme automaticky editovať stĺpec riadka tabuľky, ktorý je natiahnutý v komponente PxWebQuery. Vytvoríme si súbor AddAdresar.aspx, a do neho zapíšeme nasledujúcu definíciu komponenty PxComboBox:

Súbor AddAdresar.aspx:

```
<Prx:PxComboBox ID="cmbPravnaForma" runat="server" AddTableRow="True"  ></Prx:PxComboBox>
```

V súbore AddAdresar.aspx.cs, si zase zadefinujeme prepojenie komponenty *PxComboBox* s komponentou *PxWebQuery*, a jej prepojenie na konkrétny stĺpec tabuľky.

Súbor AddAdresar.aspx.cs:

```
protected void Page_Load(object sender, EventArgs e)
{
   cmbPravnaForma.PxDataSource = wquAdresar;
   cmbPravnaForma.FieldName = wquAdresar.Columns["idpravnaforma"].ToString();
}
```

Komponenta *PxComboBox* podporuje automatické načítanie zoznamu do svojho tela, a automatické uloženie vybranej hodnoty do databázy, pri vykonaní príkazu Post() na komponente

PxWebQuery. Na to aby došlo ku automatickému načítaniu dát do zoznamu komponenty *PxComboBox*, musíme na komponentu *PxWebQuery*, na ktorú je napojená naša komponenta *PxComboBox*, uvaliť parameter *AddParamKey*, poprípade parameter *AddParamWebQuery*.

AddParamKey slúži na prepojenie tabuľky z číselníkom. Metóda *AddParamKey* je metódou komponenty *PxWebQuery*. Filozofia Px Frameworku je taká, aby čo najviac informácii bolo definovaných na komponente *PxWebQuery*. Týmto sa vlastne šetrí programovanie a zápis pri komponentách *PxSuperGrid*, *PxEdit*, *PxComboBox* a pod. Všetky tieto komponenty si požičiavajú dáta a nastavenie z komponenty *PxWebQuery*. Preto na komponente *PxWebQuery* máme množstvo procedúr a metód, prostredníctvom ktorých definujeme rôzne zobrazenie a prepojenie medzi tabuľkami.

Taká najjednoduchšia metóda prostredníctvom, ktorej sa definuje prepojenie medzi hlavnou tabuľkou a číselníkom je metóda *AddParamKey*.

Tuto definíciu môžeme zapísať do súboru Adresar.aspx.cs.
Príklad č.1.

```
protected void Page_Load(object sender, EventArgs e)
{
  if ((!IsPostBack)&&(wquAdresar.Active==false))
  {
    string sSQLText1 = "select IDPravForm as Key, Name as Value from PravForm Order By Value";

    wquAdresar.ConnectString = "User Id=skodapeter;Password=aa;Data Source=xe;";
    wquAdresar.SQLSelect = "select idadresar , name, address, idpravnaforma, create_date from Adresar";

    wquAdresar.AddParamKey("idpravnaforma", "Key", "Value", "Value", "Key", sSQLText1);

    wquAdresar.Open();

    wquAdresar.Columns["idadresar"].Caption = "ID";
    wquAdresar.Columns["name"].Caption = "Meno Osoby";
    wquAdresar.Columns["idpravnaforma"].Caption = "Právna Forma";
    wquAdresar.Columns["address"].Caption = "Adresa";
    wquAdresar.Columns["create_date"].Caption = "Dátum založenia";
  }
  else
  {
  }
    grdAdresar.PxDataSource = wquAdresar;
    grdAdresar.DataBind();
}
```

Popis Parametrov metódy *AddParamKey*:

C# syntax:

```
public void AddParamKey(string aFieldName, string aFieldNameKey, string aFieldNameValue,
        string aFieldToView, string aFieldToDB,
        string aSQLText);
```

Popis parametrov:

aFieldName – Meno stĺpca v tabuľke na ktorý sa aplikuje metóda AddParamKey
aFieldNameKey – Meno stĺpca číselníka, ktorý reprezentuje kľúč číselníka (toto meno stĺpca sa dá zistiť z Sql príkazu, ktorý sa zadáva do parametra **aSQLText**)
aFieldNameValue – Meno stĺpca číselníka, ktorý reprezentuje hodnotu(Value) číselníka (toto meno stĺpca sa dá zistiť z Sql príkazu, ktorý sa zadáva do parametra **aSQLText**)
aFieldToView – tu sa zadáva meno stĺpca, ktorého hodnota sa bude zobrazovať v Gride(*PxSuperGrid*)
aFieldToDB - tu sa zadáva meno stĺpca, ktorého hodnota bude zapísaná do databázy
aSQLText – tu sa zadefinuje SQL príkaz pre načítanie číselníka, jednotlivé stĺpce je vhodne označiť ako Key a Value, kvôli lepšej prehľadnosti

Po uvalení parametra AddParamKey na komponentu PxWebQuery, a napojení komponenty PxComboBox, si táto komponenta pri načítaní automaticky natiahne dáta, a pri ukladaní sa automaticky uloží zvolená hodnota do databázy.

Na zarovnanie komponenty sa opäť používajú property AddTableRow, TableBegin a TableEnd, ktoré sú dobre popísané a rozobrané v časti kde je popísaná komponenta PxEdit.

Názov popiskov Caption, alebo názvy stĺpcov v gride(Title) môžete zadať centrálne, pri definícii komponenty PxWebQuery, kde hodnota v hranatých zátvorkách je názov stĺpca:

```
wquAdresar.Columns["NAME"].Caption = "Priezvisko  a Meno";
```

Zmena popisku komponenty PxComboBox (Label):

Zmenu popisku komponenty PxComboBox, okrem centrálnej definície, môžeme vykonať nasledujúcim spôsobom:

```
cmbPravnaForma.Caption = "Právna forma 2";
```

Zmena popisku At Post komponenty PxComboBox (Label):

Komponenta PxComboBox obsahuje ešte ďalší popisok, ktorý sa nachádza za komponentov TextBox. Ku tomuto popisku sa dá pristúpiť cez property *RCaption*.

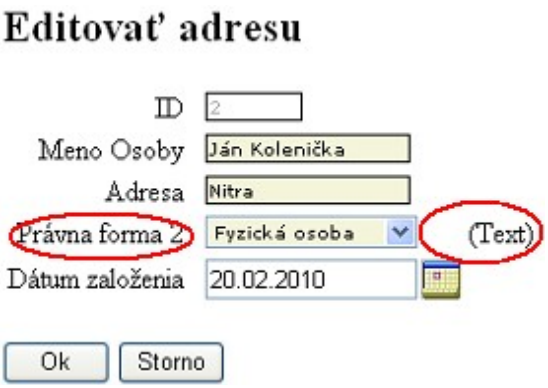
```
cmbPravnaForma.RCaption = "(Text)";
```

Výsledný tvar zdrojového kódu bude potom nasledujúci:

```
cmbPravnaForma.PxDataSource = wquAdresar;
```

```
cmbPravnaForma.FieldName = wquAdresar.Columns["IDADRESAR"].ToString();
cmbPravnaForma.Caption = "Právna forma 2";
cmbPravnaForma.RCaption = "(Text)";
```

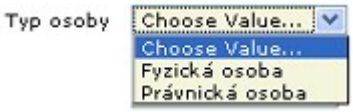
Výsledok si môžete pozrieť na obrázku:



Ak pri komponenta *PxComboBox*, pri jej ukladaní, nezvolíte žiadnu hodnotu, a necháte tam pôvodný text „Zvoľte Hodnotu“, potom sa do databázy pre daný stĺpec uloží hodnota null. Ak by Vám text „Zvoľte Hodnotu“, nevyhovoval môžete ho zmeniť cez property **ChooseValueText,** viac príklad dole**:**

```
cmbPravnaForma.ChooseValueText = "Choose Value...";
cmbPravnaForma.PxDataSource = wquAdresar;
cmbPravnaForma.FieldName = "idpravnaforma";
```

Komponenta *PxComboBox*, by vyzerala nasledovne:



1.6. PxCheckBox – komponenta, ktorá slúži na zaškrtnutie (select / unselect) hodnoty

Komponenta PxCheckBox pracuje obdobne, alebo ma podobné property, ako komponenta PxEdit. Preto nemá zmysel sa nimi podrobne zaoberať, môžete si prečítať kapitolu o komponente PxEdit. Veci, ktoré boli podrobne popísané pri komponente PxEdit, budú tuná prejdené len telegraficky, zbežne.

Pomocou komponenty PxCheckBox môžeme automaticky editovať stĺpec riadka tabuľky, ktorý je natiahnutý v komponente PxWebQuery. Vytvoríme si súbor AddAdresar.aspx, a do neho zapíšeme nasledujúcu definíciu komponenty PxComboBox:

Súbor AddAdresar.aspx:

```
<Prx:PxCheckBox ID="chkInvalid_Adress" runat="server" AddTableRow="True" ></Prx:PxCheckBox>
```

V súbore AddAdresar.aspx.cs, si zase zadefinujeme prepojenie komponenty *PxCheckBox* s komponentou *PxWebQuery*, a jej prepojenie na konkrétny stĺpec tabuľky.

Súbor AddAdresar.aspx.cs:

```
protected void Page_Load(object sender, EventArgs e)
{
    chkInvalid_Adress.PxDataSource = wquAdresar;
    chkInvalid_Adress.FieldName = wquAdresar.Columns["Invalid_Adress"].ToString();
}
```

Komponenta *PxCheckBox* podporuje automatické načítanie hodnoty z komponenty PxWebQuery, a automatické uloženie zadanej hodnoty do databázy, pri vykonaní príkazu Post() na komponente PxWebQuery. Tým že komponenta PxCheckBox umožňuje, vybrať len dva stavy, a to True, alebo False, je možne zadať na komponente PxWebQuery, ako tieto stavy budú zobrazované a zapísané do databázy. Na definíciu sa používa parameter s názvom metódy *AddParamCheck*. Metóda AddParamCheck je metódou komponenty PxWebQuery.

Popis Parametrov metódy *AddParamCheck*:

C# syntax:

```
public void AddParamCheck(string aFieldName,
                          string aCheckedValue, string aUnCheckedValue,
                          bool aEditReadOnly)
```

Popis parametrov:

aFieldName – Meno stĺpca v tabuľke na ktorý sa aplikuje metóda AddParamCheck
aCheckedValue – Hodnota reťazca, ktorou sa nahradí hodnota True, pri výbere v komponente PxCheckBox
aUnCheckedValue – Hodnota reťazca, ktorou sa nahradí hodnota False, pri výbere v komponente PxCheckBox
aEditReadOnly - Keď bude Grid v stave edit, pri zadanej hodnote True bude povolená editácia

Po uvalení parametra AddParamCheck na komponentu PxWebQuery, a napojení komponenty PxCheckBox, si táto komponenta pri načítaní automaticky vie interpretovať načítané dáta z databázy, a pri ukladaní sa automaticky uloží zadefinovaná hodnota reťazca do databázy.

V našom príklade, bude definícia parametra AddParamCheck vyzerať nasledovne:

```
wquAdresar.AddParamCheck("invalid_adress", "A", "N", false);
```

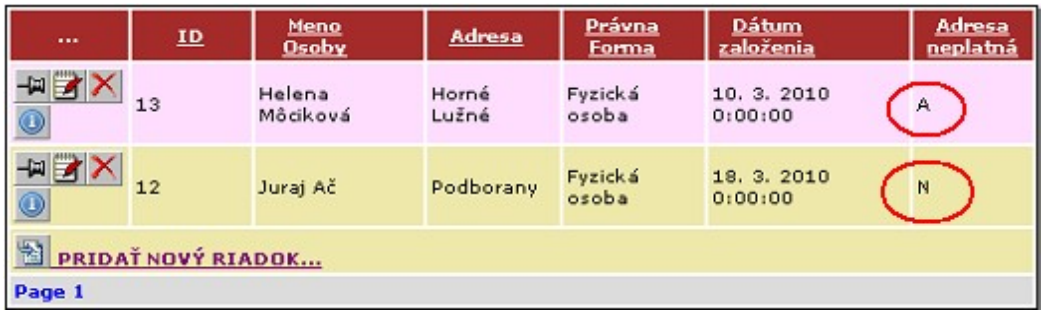
Tuto definíciu potom môžeme zapísať do súboru Adresar.aspx.cs, a bude vyzerať nasledovne.

Príklad 05_PxCheckBox.

```
protected void Page_Load(object sender, EventArgs e)
{
   if ((!IsPostBack)&&(wquAdresar.Active==false))
   {
      string sSQLText1 = "select IDPravForm as Key, Name as Value from PravForm Order By Value";

      wquAdresar.ConnectString = "User Id=skodapeter;Password=aa;Data Source=xe;";
      wquAdresar.SQLSelect = "select idadresar , name, address, idpravnaforma, create_date, invalid_adress from
                                 Adresar02";

      wquAdresar.AddParamKey("idpravnaforma", "Key", "Value", "Value", "Key", sSQLText1);
      wquAdresar.AddParamCheck("invalid_adress", "A", "N", false);
      wquAdresar.Open();

      wquAdresar.Columns["idadresar"].Caption = "ID";
      wquAdresar.Columns["name"].Caption = "Meno Osoby";
      wquAdresar.Columns["idpravnaforma"].Caption = "Právna Forma";
      wquAdresar.Columns["address"].Caption = "Adresa";
      wquAdresar.Columns["create_date"].Caption = "Dátum založenia";
      wquAdresar.Columns["invalid_adress"].Caption = "Adresa neplatná";

   }
   else
   {
   }
      grdAdresar.PxDataSource = wquAdresar;
      grdAdresar.DataBind();
}
```

Po zapísaný, respektíve uvalení parametra AddParamCheck na komponentu PxWebQuery, sa nám v komponente PxSuperGride budú zobrazovať hodnoty, namiesto hodnoty True to bude hodnota „A“ a namiesto hodnoty False to bude hodnota „N“. Podobne bude vykonaný aj zápis do databázy.

Viac obrázok dole:



...	ID	Meno Osoby	Adresa	Právna Forma	Dátum založenia	Adresa neplatná
	13	Helena Môcíková	Horné Lužné	Fyzická osoba	10. 3. 2010 0:00:00	A
	12	Juraj Ač	Podborany	Fyzická osoba	18. 3. 2010 0:00:00	N

PRIDAŤ NOVÝ RIADOK...

Page 1

Vo formuláry AddAdresar.aspx to bude vyzerať nasledovne:

Na zarovnanie komponenty sa opäť používajú property AddTableRow, TableBegin a TableEnd, ktoré sú dobre popísané a rozobrané v časti kde je popísaná komponenta PxEdit.

Názov popiskov Caption, alebo názvy stĺpcov v gride(Title) môžete zadať centrálne, pri definícii komponenty PxWebQuery, kde hodnota v hranatých zátvorkách je názov stĺpca:

```
wquAdresar.Columns["invalid_adress"].Caption = "Adresa neplatná";
```

Zmena popisku komponenty PxCheckBox (Label):

Zmenu popisku komponenty PxCheckBox, okrem centrálnej definície, môžeme vykonať nasledujúcim spôsobom:

```
chkInvalid_Adress.Text = "Adresa neplatná 2";
```

Pokiaľ chceme zmeniť text v komponente PxCheckBox, musíme zadať reťazec do property Text.

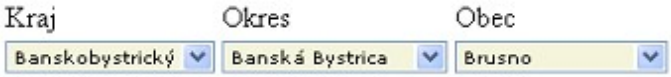
Výsledný tvar zdrojového kódu bude potom nasledujúci:

```
chkInvalid_Adress.PxDataSource = wquAdresar;
chkInvalid_Adress.FieldName = wquAdresar.Columns["Invalid_Adress"].ToString();
chkInvalid_Adress.Text = "Adresa neplatná 2";
```

Výsledok si môžete pozrieť na obrázku:

1.7. PxFlyComboBox – séria comboboxov, zapojených za sebou, vhodný pre prácu zo štruktúrovanými dátami (napr. výber kategórii a podkategórii)

Komponenta PxFlyComboBox, slúži na výber hodnoty, pri štruktúrovaných zoznamoch, najčastejšie použitie býva, keď sa nejaký objekt začleňuje do nejakej kategórie, a podkategórie. Komponenta PxFlyComboBox sa môže skladať z niekoľkých comboboxov(komponenty DropDownList), prepojených vzájomne medzi sebou. Týchto comboboxov môže byť ľubovolný počet 2 až n. Viac obrázok:



Na obrázku máme vyber názvu obce. Pretože, obcí na Slovensku máme skoro 3000, bolo by dosť obtiažne vyberať z takéhoto zoznamu. Tu nám uľahčí prácu, komponenta PxFlyComboBox. Pri výbere konkrétneho krája, sa nám do comboboxu okres, vyfiltrujú len tie okresy, za daný kraj. Keď si zvolíme konkrétny okres, tak sa nám do comboboxu obcí vyfiltrujú, len obce za daný zvolení okres.
Naprogramovať, takéto niečo, bohužiaľ nie je jednoduché, ale PxFramework Vám poskytuje rýchle a lacné riešenie.

Na to, aby Vám komponenta PxFlyComboBox, správne fungovala musíte uvaliť parameter AddParamFlyComboBox, na danú komponentu PxWebQuery, ktorá bude zobrazovať alebo pracovať s takto štruktúrovanými dátami. Definícia parametra AddParamFlyComboBox vyzerá nasledovne:

```
wquAdresar.AddParamFlyComboBox("miesto", "idkraj;idokres;idobec",
                               "idkraj;idokres;idobec", "kraj;okres;obec",
                               "Obec;Okres", "wquObec", "mvStandard");
```

Popis Parametrov metódy *AddParamFlyComboBox*:

C# syntax:

```
public void AddParamFlyComboBox(string aFieldName, string aKeyFields,
                                string aKeyDetailFields, string aValueFields,
                                string aFieldToView, string aPxWebQueryName,
                                string aModeToView)
```

Popis parametrov:

aFieldName – Meno stĺpca, ktorý sa vytvorí ako nový stĺpec v tabuľke na ktorý sa aplikuje metóda AddParamFlyComboBox, a na ktorý sa potom napája komponenta PxFlyComboBox (toto meno stĺpca musí byť jedinečne, v tabuľke nesmie existovať stĺpec s týmto menom).
aKeyFields – Hodnota reťazca, do ktorej sa definujú kľúče (Key) stĺpcov hlavnej tabuľky, do ktorých sa pri editácii komponentou PxFlyComboBox zapisujú Idečka, v našom prípade IdKraja, IdOkresu a IdObce. Tieto stĺpce sú zapísané v poradí, jeden za druhým, ako sa majú zobraziť v komponente PxFlyComboBox.

Jednotlivé stĺpce sú oddelené bodkočiarkou.
aKeyDetailFields – Hodnota reťazca, do ktorej sa definujú kľúče (Key) stĺpcov tabuľky číselníka, z ktorých sa pri editácii komponentou PxFlyComboBox vyberajú Idečka, a zapisujú do hlavnej tabuľky. V našom prípade sa jedná o stĺpce IdKraja, IdOkresu a IdObce. Tieto stĺpce sú zapísané v poradí, jeden za druhým, ako sa majú zobraziť v komponente PxFlyComboBox. Jednotlivé stĺpce sú oddelené bodkočiarkou.
aValueFields - Hodnota reťazca, do ktorej sa definujú hodnoty (Value) stĺpcov tabuľky číselníka, ktoré sa pri editácii komponentou PxFlyComboBox zobrazujú v tejto komponente. V našom prípade sa jedná o stĺpce Kraja, Okresu a Obce. Tieto stĺpce sú zapísané v poradí, jeden za druhým, ako sa majú zobraziť v komponente PxFlyComboBox. Jednotlivé stĺpce sú oddelené bodkočiarkou.
aFieldToView – Hodnota reťazca, ktorou sa definujú mená stĺpcov, ktoré sú zobrazované v komponente PxSuperGrid, pod názvom stĺpca aFieldName. Vypis, alebo výčet týchto stĺpcov nemusí byť kompletný. Tieto stĺpce sú zapísané v poradí, jeden za druhým, ako sa majú zobraziť v komponente PxSuperGrid. Jednotlivé názvy stĺpcov sú oddelené bodkočiarkou.
aPxWebQueryName – Názov komponenty PxWebQuery, v ktorej je natiahnutý číselník Kraja, Okresu a Obce. Tento číselník je vytvorený zlúčením troch číselníkov a to v našom prípade číselníkov Ckraj, Cokres, CObec.
aModeToView – Týmto reťazcom sa definuje zobrazenie dát v komponente PxSuperGrid, v našom prípade stĺpca zadefinovaného v položke aFieldName. Existujúce módy zobrazenia sú nasledovné: **mvValueBR, mvNameAndBR a mvNormal**.

Pri zadaní reťazca "Obec;Okres" do položky FieldToView, nastavenom móde zobrazenia **mvNormal**, bude zobrazenie v komponente PxSuperGrid vyzerať nasledovne:

Pri zadaní reťazca "Obec;Okres" do položky FieldToView, nastavenom móde zobrazenia **mvValueBR**, bude zobrazenie v komponente PxSuperGrid vyzerať nasledovne:

Pri zadaní reťazca "Obec;Okres" do položky FieldToView, nastavenom móde zobrazenia **mvNameAndBR**, bude zobrazenie v komponente PxSuperGrid vyzerať nasledovne:

Pri zadaní reťazca "Obec;Okres;Kraj" do položky FieldToView, nastavenom móde zobrazenia **mvNameAndBR**, bude zobrazenie v komponente PxSuperGrid vyzerať nasledovne:

Celkové zobrazenie komponenty PxSuperGrid bude vyzerať nasledovne:

Ale vráťme sa ku konkrétnemu programovaniu s komponentov PxFlyComboBox, nášmu príkladu adresára osôb a firiem.

Na to aby sme mohli uvaliť parameter AddParamFlyComboBox na komponentu wquAdresar, musíme vytvoriť novu komponentu PxWebQuery, a tu nazveme wquObec. V súbore Adreasar.aspx doplníme túto novú komponentu, a definícia bude vyzerať nasledovne:

```
<Prx:PxWebQuery ID="wquObec" runat="server" Value="wquObec"/>
<Prx:PxWebQuery ID="wquAdresar" runat="server" Value="wquAdresar"/>
```

Nezabudnime komponentu wquObec, zadefinovať pred komponentou wquAdresar.

Ďalej v súbore Adreasar.aspx.cs načítame dáta do komponenty wquObec, a potom urobíme prepojenie tejto komponenty s komponentou wquAdresar, cez parameter AddParamFlyComboBox. Pri definovaný komponenty wquObec, musíme vytvoriť špeciálny SQL select, do ktorého zlúčime tri tabuľky, a to tabuľky Ckraj, Cokres, CObec. Tento SQL príkaz vyzerá nasledovne:

```
select COBEC.ID_OBEC as IDOBEC, COBEC.NAZOV_OBEC as OBEC,
   COBEC.ID_OKRES as IDOKRES, COKRES.NAZOV_OKRES as OKRES,
   COBEC.ID_KRAJ as IDKRAJ,CKRAJ.NAZOV_KRAJ as KRAJ
from
   COBEC,COKRES,CKRAJ
where COBEC.ID_OKRES=COKRES.ID_OKRES and
   COBEC.ID_KRAJ=CKRAJ.ID_KRAJ
```

Výsledkom takéhoto SQL príkazu je tabuľka, viz. na obrázku dole:

IDOBEC	OBEC	IDOKRES	OKRES	IDKRAJ	KRAJ
1	Vadovce	19	Nové Mesto nad Váhom	3	Trenciansky
2	Velká Hradná	24	Trenčín	3	Trenciansky
3	Velké Bierovce	24	Trenčín	3	Trenciansky
4	Višnové	19	Nové Mesto nad Váhom	3	Trenciansky
5	Trnava	15	Trnava	2	Trnavský
6	Bínovce	15	Trnava	2	Trnavský
7	Bohdanovce n...	15	Trnava	2	Trnavský

Z takouto tabuľkou, komponenta PxFlyComboBox vie pracovať pomerne celkom dobre.
A tu v ďalšom máte výpis zdrojového kódu súboru Adresar.aspx.cs, pre naplnenie dát komponenty wquObec, a jej prepojenie z komponentou wquAdresar, cez parameter AddParamFlyComboBox.

```
protected void Page_Load(object sender, EventArgs e)
{
   if ((!IsPostBack) && (wquObec.Active == false))
   {
      wquObec.ConnectString = "User Id=skodapeter;Password=aa;Data Source=xe;";
      wquObec.SQLSelect = @"select COBEC.ID_OBEC as IDOBEC, COBEC.NAZOV_OBEC as OBEC,
                  COBEC.ID_OKRES as IDOKRES, COKRES.NAZOV_OKRES as OKRES,
                  COBEC.ID_KRAJ as IDKRAJ,CKRAJ.NAZOV_KRAJ as KRAJ
                  from
                     COBEC,COKRES,CKRAJ
                  where COBEC.ID_OKRES=COKRES.ID_OKRES and
                     COBEC.ID_KRAJ=CKRAJ.ID_KRAJ";

      wquObec.Open();

      wquObec.Columns["IDOBEC"].Caption = "Id Obec";
      wquObec.Columns["OBEC"].Caption = "Obec";
      wquObec.Columns["IDOKRES"].Caption = "Id Okres";
      wquObec.Columns["OKRES"].Caption = "Okres";
      wquObec.Columns["IDKRAJ"].Caption = "Id Kraj";
      wquObec.Columns["KRAJ"].Caption = "Kraj";
   }

   if ((!IsPostBack)&&(wquAdresar.Active==false))
   {
      string sSQLText1 = "select IDPravForm as Key, Name as Value from PravForm Order By Value";

      wquAdresar.ConnectString = "User Id=skodapeter;Password=aa;Data Source=xe;";
      wquAdresar.SQLSelect = "select idadresar , name, address, idpravnaforma, create_date, invalid_adress, idkraj,
idokres, idobec from Adresar03";

      wquAdresar.AddParamKey("idpravnaforma", "Key", "Value", "Value", "Key", sSQLText1);
      wquAdresar.AddParamCheck("invalid_adress", "A", "N", false);
      wquAdresar.AddParamFlyComboBox("miesto", "idkraj;idokres;idobec",
```

```
                                "idkraj;idokres;idobec", "kraj;okres;obec",
                                "Obec;Okres;Kraj", "wquObec", "mvNameAndBR");

        wquAdresar.Open();

        wquAdresar.Columns["idadresar"].Caption = "ID";
        wquAdresar.Columns["name"].Caption = "Meno Osoby";
        wquAdresar.Columns["idpravnaforma"].Caption = "Právna Forma";
        wquAdresar.Columns["address"].Caption = "Ulica";
        wquAdresar.Columns["create_date"].Caption = "Dátum založenia";
        wquAdresar.Columns["invalid_adress"].Caption = "Adresa neplatná";
        wquAdresar.Columns["miesto"].Caption = "Obec, Okres";

    }
    else
    {
    }
        grdAdresar.PxDataSource = wquAdresar;
        grdAdresar.DataBind();
}
```

Pokiaľ chceme vo formulári AddAdresar.aspx, chceme použiť komponentu PxFlyComboBox na editáciu, tak ju v tomto súbore zadefinujeme nasledovne:

```
<Prx:PxWebQuery ID="wquObec" runat="server" Value="wquObec"/>
<Prx:PxWebQuery ID="wquAdresar" runat="server" Value="wquAdresar"/>

<Prx:PxEdit ID="edtIDADRESAR" runat="server" AddTableRow="True" TableBegin="True" ></Prx:PxEdit>
<Prx:PxEdit ID="edtName" runat="server" AddTableRow="True"  ></Prx:PxEdit>
<Prx:PxComboBox ID="cmbPravnaForma" runat="server" AddTableRow="True"  ></Prx:PxComboBox>
<Prx:PxJSDatePicker ID="edtCREATE_DATE" runat="server" AddTableRow="True"  ></Prx:PxJSDatePicker>
<Prx:PxCheckBox ID="chkInvalid_Adress" runat="server" AddTableRow="True" ></Prx:PxCheckBox>
<Prx:PxEdit ID="edtADDRESS" runat="server" AddTableRow="True" TableEnd="True"></Prx:PxEdit>
<Prx:PxFlyComboBox ID="fcmbMiesto" runat="server"></Prx:PxFlyComboBox>
```

V súbore AddAdresar.aspx.cs prepojíme komponentu PxFlyComboBox s komponentou wquAdresar, toto prepojenie by malo vyzerať nasledovne:

```
fcmbMiesto.PxDataSource = wquAdresar;
fcmbMiesto.FieldName = wquAdresar.Columns["miesto"].ColumnName;
```

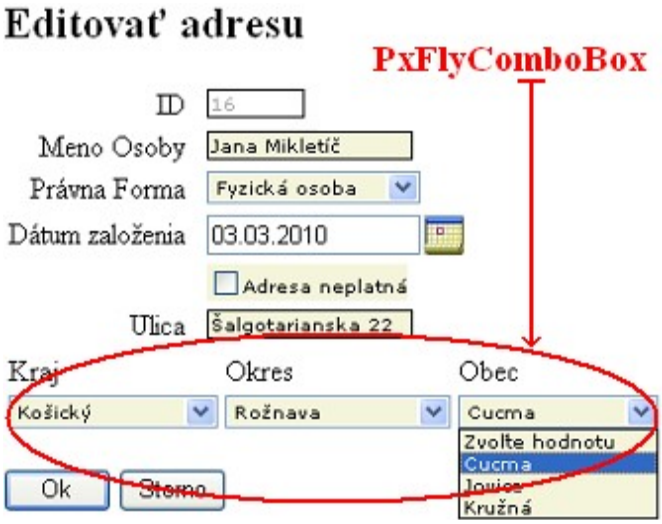
Pri definícii FieldName, treba použiť property ColumnName, a nie ToString(), lebo komponenta PxFlyComboBox sa nám vo formuláry nezobrazí.

Komponenta PxFlyComboBox nepodporuje zarovnanie a property AddTableRow, TableBegin a TableEnd, ktoré podporujú ostatné komponenty PxEdit a pod.

Pri definícii v súbore AddAdresar.aspx, nesmieme ešte zabudnúť okrem definovania riadku komponenty PxFlyComboBox, definovať komponentu wquObec, lebo ak túto zabudneme definovať, komponenta PxFlyComboBox sa nám vo formuláry AddAdresar.aspx nezobrazí.

```
<Prx:PxWebQuery ID="wquObec" runat="server" Value="wquObec"/>
<Prx:PxWebQuery ID="wquAdresar" runat="server" Value="wquAdresar"/>
```

Ak sme nastavili všetko tak, ako sme mali výsledný formulár AddAdresar.aspx, bude vyzerať nasledovne:



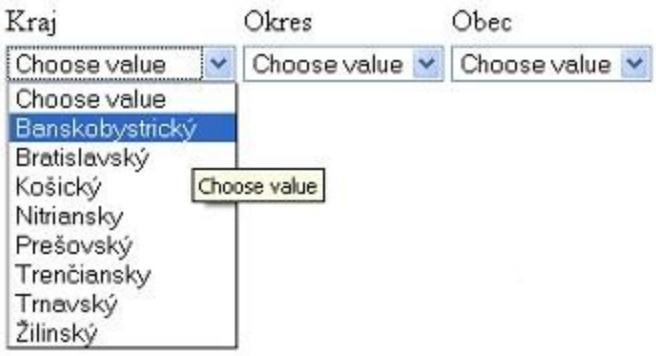
Popisky v Gride(Title) a popisky v komponente PxFlyComboBox (Caption), sa dajú definovať centrálne pri definícii komponenty wquObec:

```
wquObec.Columns["IDOBEC"].Caption = "Id Obec";
wquObec.Columns["OBEC"].Caption = "Obec";
wquObec.Columns["IDOKRES"].Caption = "Id Okres";
wquObec.Columns["OKRES"].Caption = "Okres";
wquObec.Columns["IDKRAJ"].Caption = "Id Kraj";
wquObec.Columns["KRAJ"].Caption = "Kraj";
```

Ak pri komponenta *PxFlyComboBox*, pri jej ukladaní, nezvolíte žiadnu hodnotu, a necháte tam pôvodný text „Zvoľte Hodnotu“, potom sa do databázy pre daný stĺpec uloží hodnota null. Ak by Vám text „Zvoľte Hodnotu“, nevyhovoval môžete ho zmeniť cez property **ChooseValueText,** viac príklad dole**:**

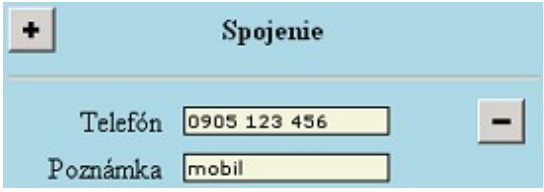
```
fcmbMiesto.ChooseValueText = "Choose Value";
fcmbMiesto.PxDataSource = wquAdresar;
fcmbMiesto.FieldName = wquAdresar.Columns["miesto"].ColumnName;
```

Komponenta *PxFlyComboBox*, by vyzerala nasledovne:



1.8. PxGreatRepeater – komponenta pre zadávanie dát, ktoré sa svojou štruktúrou opakujú, maximálny počet hodnôt je limitovaný

Komponenta PxGreatRepeater slúži na zadávanie dát, ktoré sa svojou štruktúrou cyklicky opakujú. Taký jednoduchý príklad, kde sa dáta opakujú, môžeme nájsť pri osobe, ktorá má telefónne číslo, respektíve môže mať týchto telefónnych čísel viac. Dole je zobrazený ovládací prvok PxGreatRepeater, na zadanie tel. čísla:



Samozrejme cez ovládací prvok PxGreatRepeater, môžeme zadať viac telefónnych čísiel, viac obrázok dolu:

Tlačítkom plus „+“ pridávame nové okienka pre zadanie telefónneho čísla, tlačítkom „-“ môžeme mazať už existujúce telefónne čísla.
A teraz ako to funguje v praxi. Do tabuľky adresár si doplníme nasledujúce stĺpce:

```
Tel1  VARCHAR2(50),
Tel_Pozn1  VARCHAR2(50),
Tel2  VARCHAR2(50),
Tel_Pozn2  VARCHAR2(50),
Tel3  VARCHAR2(50),
Tel_Pozn3  VARCHAR2(50)
```

Ako vidíte do tabuľky adresár sme zadali položku telefón(tel1, tel2, tel3) tri krát. Teda pri zadávaný telefónneho čísla a jeho poznámky budeme limitovaný, môžeme zadať maximálne tri telefónne čísla, aj s poznámkou. Zadávanie a mazanie, môžeme manažovať tlačítkom plus „+“ a tlačítkami mínus „-“.

Na to aby sme komponentu PxGreatRepeater pripojili ku komponente PxWebQuery, potrebujeme uvaliť parameter *AddParamGreatWebQuery* na komponentu PxWebQuery.

Na to, aby Vám komponenta PxGreatRepeater, správne fungovala musíte uvaliť parameter AddParamGreatWebQuery, na danú komponentu PxWebQuery, ktorá bude zobrazovať alebo pracovať s takto štruktúrovanými dátami. Definícia parametra AddParamGreatWebQuery vyzerá nasledovne:

```
wquAdresar.AddParamGreatWebQuery("Telefon", "Tel1;Tel_Pozn1", "Tel1 - Tel_Pozn1", 3, "mvNormal");
```

Popis Parametrov metody *AddParamGreatWebQuery*:

C# syntax:

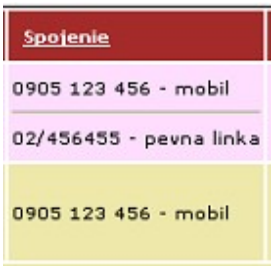
```
public void AddParamGreatWebQuery(string aFieldName,
                                  string eEditAllFieldNames,
                                  string ExpresionToView,
                                  Int16 aMaxRepeatCount,
                                  string ModeToView)
```

Popis parametrov:

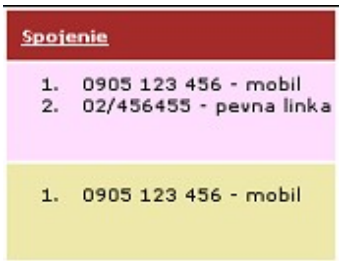
aFieldName – Meno stĺpca, ktorý sa vytvorí ako nový stĺpec v tabuľke na ktorý sa aplikuje metóda AddParamGreatWebQuery, a na ktorý sa potom napája komponenta PxGreatRepeater (toto meno stĺpca musí byť jedinečne, v tabuľke nesmie existovať stĺpec s týmto menom).
eEditAllFieldNames – Tu sa vypíšu všetky mená stĺpcov, ktoré sa budú editovať v komponente PxGreatRepeater, presne v poradí v akom budú zoradené v komponente PxGreatRepeater. Jednotlivé stĺpce sú oddelené bodkočiarkou. Tieto mená stĺpcov budú pozostávať zo stĺpcov ktoré sa cyklicky opakujú, t.j. Tel1..až..Tel3, a vždycky sa zadá meno stĺpca, ktorý sa začína jednotkou.
ExpresionToView – Tu sa vypíšu mená všetkých stĺpcov, ktoré sa majú zobrazovať v Gride. Tieto mena môžu byť medzi sebou oddelene čiarkou alebo pomlčkou, ktoré sa zobrazia pri zobrazený v Gride(PxSuperGrid).
aMaxRepeatCount – Touto hodnotou sa stanovuje maximálny počet telefónnych čísel, ktoré je možné zadať prostredníctvom komponenty PxGreatRepeater. Ak sme do databázy prirobili stĺpce tel1, tel2, tel3 potom hodnota tejto premennej bude 3.
aModeToView – Týmto reťazcom sa definuje zobrazenie dát v komponente PxSuperGrid, v našom prípade stĺpca zadefinovaného v položke aFieldName. Existujúce módy zobrazenia sú nasledovné: **mvBreakHR, mvLineNumber a mvNormal**.

Pri zadaní reťazca "Tel1 - Tel_Pozn1" do položky ExpresionToView, nastavenom móde zobrazenia **mvNormal**, bude zobrazenie v komponente PxSuperGrid vyzerať nasledovne:

Pri zadaní reťazca "Tel1 - Tel_Pozn1" do položky ExpresionToView, nastavenom móde zobrazenia **mvBreakHR**, bude zobrazenie v komponente PxSuperGrid vyzerať nasledovne:



Pri zadaní reťazca "Tel1 - Tel_Pozn1" do položky ExpresionToView, nastavenom móde zobrazenia **mvLineNumber**, bude zobrazenie v komponente PxSuperGrid vyzerať nasledovne:



Na to aby sme zredukovali počet zobrazených stĺpcov v Gride použijeme property VisibleFields komponenty PxSuperGrid, viac ukážka dole:

```
grdAdresar.VisibleFields = "name;idpravnaforma;address;miesto;telefon";
```

Do property VisibleFields sme priradili názvy stĺpcov, oddelených bodkočiarkou, ktoré sa majú zobraziť pri zobrazený v komponente PxSuperGrid-u. Ak v property VisibleFields, nie je priradené žiadna hodnota, v komponente PxSuperGrid-u sa nám zobrazia všetky stĺpce(fields). Samozrejme všetky stĺpce sa nám v komponente PxSuperGrid zobrazia presne v tom poradí, v akom poradí sme ich zadali do property VisibleFields.

Prostredníctvom property VisibleFields, môžeme zadefinovať taktiež dĺžku stĺpca, tým, že za každým stĺpcom, urobíme dvojbotku, a za ňou zadáme číslo v pixeloch, ktoré má definovať dĺžku stĺpca. Viac príklad dole:

```
grdAdresar.VisibleFields = "idadresar:50;name;idpravnaforma;address;miesto:100;telefon:150";
```

Property VisibleFields, na to aby správne fungovala, musíme byť definovaná pred pripojením a bindovaním na komponentu PxWebQuery.

```
grdAdresar.VisibleFields = "name;idpravnaforma;address;miesto;telefon";
grdAdresar.PxDataSource = wquAdresar;
grdAdresar.DataBind();
```

Výsledný PxSuperGrid bude vyzerať nasledovne:

...	Meno Osoby	Ulica	Právna Forma	Spojenie	Obec, Okres
	Peter Tkáč	Kolkareň 13/18	Fyzická osoba	1. 0905 123 456 - mobil 2. 02/456455 - pevna linka	Obec: **Čučma** Okres: **Rožňava** Kraj: **Košický**
	Helena Môciková	Šalgotarianska 22	Fyzická osoba	1. 0905 123 456 - mobil	Obec: **Dvorníky - Včeláre** Okres: **Košice - okolie** Kraj: **Košický**

PRIDAŤ NOVÝ RIADOK...

Page 1

A tu v ďalšom máte výpis zdrojového kódu súboru Adresar.aspx.cs, pre aplikáciu parametra AddParamGreatWebQuery na komponentu wquAdresar. V príklade sú uvedené zadania dvoch parametrov, a to telefónnu a faxu:

```
if ((!IsPostBack)&&(wquAdresar.Active==false))
  {
    string sSQLText1 = "select IDPravForm as Key, Name as Value from PravForm Order By Value";

    wquAdresar.ConnectString = "User Id=skodapeter;Password=aa;Data Source=xe;";
    wquAdresar.SQLSelect = @"select idadresar , name, address, idpravnaforma,
                    create_date, invalid_adress, idkraj, idokres,
                    idobec, tel1, tel_pozn1, tel2, tel_pozn2,
                    tel3, tel_pozn3, fax1, fax_pozn1,
                    fax2, fax_pozn2 from Adresar04";

    wquAdresar.AddParamKey("idpravnaforma", "Key", "Value", "Value", "Key", sSQLText1);
    wquAdresar.AddParamCheck("invalid_adress", "A", "N", false);
    wquAdresar.AddParamFlyComboBox("miesto", "idkraj;idokres;idobec", "idkraj;idokres;idobec",
                                   "kraj;okres;obec",
                                   "Obec;Okres;Kraj", "wquObec", "mvNameAndBR");

    wquAdresar.AddParamGreatWebQuery("Telefon", "Tel1;Tel_Pozn1", "Tel1 - Tel_Pozn1", 3,
                                     "mvLineNumber");
    wquAdresar.AddParamGreatWebQuery("Fax", "Fax1;Fax_Pozn1", "Fax1 - Fax_Pozn1", 2, "");

    wquAdresar.Open();

    wquAdresar.Columns["idadresar"].Caption = "ID";
    wquAdresar.Columns["name"].Caption = "Meno Osoby";
    wquAdresar.Columns["idpravnaforma"].Caption = "Právna Forma";
    wquAdresar.Columns["address"].Caption = "Ulica";
    wquAdresar.Columns["create_date"].Caption = "Dátum založenia";
    wquAdresar.Columns["invalid_adress"].Caption = "Adresa neplatná";
    wquAdresar.Columns["miesto"].Caption = "Obec, Okres";

    wquAdresar.Columns["tel1"].Caption = "Telefón";
    wquAdresar.Columns["tel_pozn1"].Caption = "Poznámka";
    wquAdresar.Columns["tel2"].Caption = "Telefón";
    wquAdresar.Columns["tel_pozn2"].Caption = "Poznámka";
    wquAdresar.Columns["tel3"].Caption = "Telefón";
    wquAdresar.Columns["tel_pozn3"].Caption = "Poznámka";

    wquAdresar.Columns["fax1"].Caption = "Fax";
    wquAdresar.Columns["fax_pozn1"].Caption = "Poznámka";
    wquAdresar.Columns["fax2"].Caption = "Fax";
```

```
        wquAdresar.Columns["fax_pozn2"].Caption = "Poznámka";

        wquAdresar.Columns["telefon"].Caption = "Spojenie";

    }
    else
    {
    }
    grdAdresar.VisibleFields = "name;idpravnaforma;address;miesto;telefon";
    grdAdresar.PxDataSource = wquAdresar;
    grdAdresar.DataBind();
    }
```

Pokiaľ chceme vo formulári AddAdresar.aspx použiť komponentu PxGreatRepeater na editáciu, tak ju v tomto súbore zadefinujeme nasledovne:

```
<Prx:PxWebQuery ID="wquObec" runat="server" Value="wquObec"/>
<Prx:PxWebQuery ID="wquAdresar" runat="server" Value="wquAdresar"/>

<Prx:PxEdit ID="edtIDADRESAR" runat="server" AddTableRow="True" TableBegin="True" ></Prx:PxEdit>
<Prx:PxEdit ID="edtName" runat="server" AddTableRow="True"  ></Prx:PxEdit>
<Prx:PxComboBox ID="cmbPravnaForma" runat="server" AddTableRow="True"  ></Prx:PxComboBox>
<Prx:PxJSDatePicker ID="edtCREATE_DATE" runat="server" AddTableRow="True"  ></Prx:PxJSDatePicker>
<Prx:PxCheckBox ID="chkInvalid_Adress" runat="server" AddTableRow="True" ></Prx:PxCheckBox>
<Prx:PxEdit ID="edtADDRESS" runat="server" AddTableRow="True" TableEnd="True"></Prx:PxEdit>
<Prx:PxFlyComboBox ID="fcmbMiesto" runat="server"></Prx:PxFlyComboBox>
<br />
<Prx:PxGreatRepeater ID="gtrTelefon" runat="server"></Prx:PxGreatRepeater>
<br />
<Prx:PxGreatRepeater ID="gtrFax" runat="server"></Prx:PxGreatRepeater>
```

V súbore AddAdresar.aspx.cs prepojíme komponentu PxGreatRepeater s komponentou wquAdresar, toto prepojenie by malo vyzerať nasledovne:

```
gtrTelefon.PxDataSource = wquAdresar;
gtrTelefon.FieldName = wquAdresar.Columns["telefon"].ColumnName;

gtrFax.PxDataSource = wquAdresar;
gtrFax.FieldName = wquAdresar.Columns["fax"].ColumnName;
```

Pri definícii FieldName, treba použiť property ColumnName, a nie ToString(), lebo komponenta PxGreatRepeater sa nám vo formuláry nezobrazí.

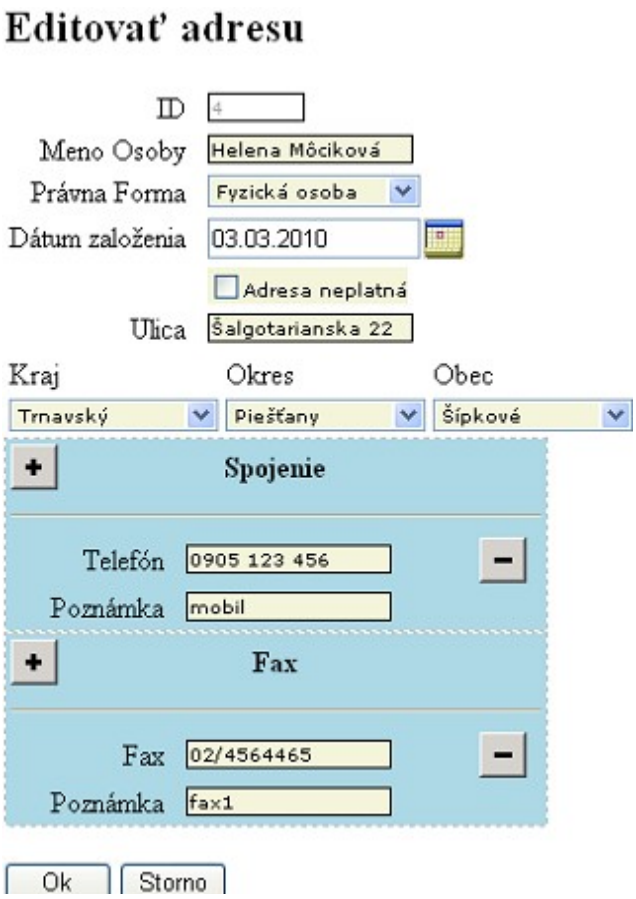
Komponenta PxGreatRepeater nepodporuje zarovnanie a property AddTableRow, TableBegin a TableEnd, ktoré podporujú ostatné komponenty PxEdit apod., a preto treba použiť tág <BR />.

Popisky v Gride(Title) a popisky v komponente PxGreatRepeater (Caption), sa dajú definovať centrálne pri definícii komponenty wquAdresar:

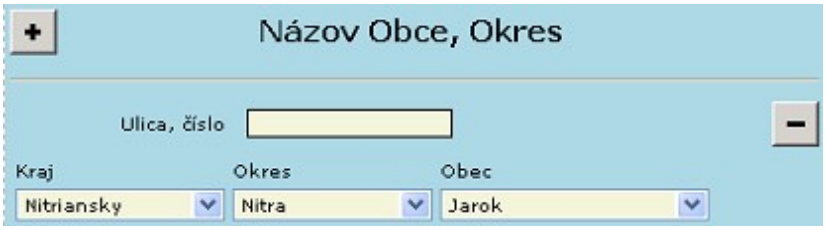
```
wquAdresar.Columns["tel1"].Caption = "Telefón";
wquAdresar.Columns["tel_pozn1"].Caption = "Poznámka";
wquAdresar.Columns["tel2"].Caption = "Telefón";
```

```
wquAdresar.Columns["tel_pozn2"].Caption = "Poznámka";
wquAdresar.Columns["tel3"].Caption = "Telefón";
wquAdresar.Columns["tel_pozn3"].Caption = "Poznámka";

wquAdresar.Columns["fax1"].Caption = "Fax";
wquAdresar.Columns["fax_pozn1"].Caption = "Poznámka";
wquAdresar.Columns["fax2"].Caption = "Fax";
wquAdresar.Columns["fax_pozn2"].Caption = "Poznámka";

wquAdresar.Columns["telefon"].Caption = "Spojenie";
```

Ak sme nastavili všetko tak, ako sme mali, výsledný formulár AddAdresar.aspx, bude vyzerať nasledovne:



V komponente PxGreatRepeater sú integrované komponenty PxComboBox, PxFlyComboBox, PxJSDatePicker, a pri uvalený daného parametra (AddParamKey, AddParamWebQuery, AddParamFlyComboBox a pod.)na daný stĺpec v komponente PxWebQuery, sa jednotlivé komponenty automaticky aktivujú v komponente PxGreatRepeater. Viac uvidíte na obrázkoch dole:



1.9. PxDbNavigator – komponenta pre prácu z komponentov PxWebQuery, posúvanie kurzora riadku a pod.

Komponenta PxDbNavigator slúži na posúvanie, respektíve na zmenu pozície kurzora, býva doplnkom ku komponente PxSuperGrid. Komponenta PxDbNavigator sa pripája na komponentu PxWebQuery, a pri stlačený príslušného tlačítka, sa prevedená zmena zobrazí aj v komponente PxSuperGrid, posunutým kurzora, alebo nejakou inou operáciou. Komponenta PxDbNavigator vyzerá nasledovne:

Pokiaľ chceme pracovať s komponentou PxDbNavigator, musíme si ju v súbore Adresar.aspx zadefinovať nasledovne:

```
<Prx:PxDbNavigator ID="dbnAdresar" runat="server" />
```

V súbore Adresar.aspx.cs pripojíme komponentu PxDbNavigator na komponentu PxWebQuery nasledovným spôsobom:

```
dbnAdresar.PxDataSource = wquAdresar;
```

Zobrazenie tlačítok môžeme ovládať cez property *PxVisibleButtons*, nasledovným spôsobom:

```
dbnAdresar.PxVisibleButtons = "FRNLIEDPC";
```

Tu v ďalšom je popísaná legenda, podľa ktorej sa zobrazujú jednotlivé tlačítka, pokiaľ je do property PxVisibleButtons zapísané konkrétne písmeno, tak sa v komponente PxDbNavigator, zobrazí príslušné tlačítko, ktoré

zodpovedá tomuto písmenu:

F – (First) presun kurzora na prvý riadok v komponente PxWebQuery
R – (Prior) presun kurzora na predchádzajúci riadok v komponente PxWebQuery
N – (Next) presun kurzora na nasledujúci riadok v komponente PxWebQuery
L – (Last) presun kurzora na posledný riadok v komponente PxWebQuery
I – (Insert) uvedenie komponenty PxWebQuery do stavu vkladania nového riadku
E– (Edit) uvedenie komponenty PxWebQuery do stavu editácie, zmeny aktuálneho riadku
D – (Delete) zmazanie aktuálneho riadku v komponente PxWebQuery
P – (Post) uloženie zmien dát po editácii, alebo vkladaní nového riadku komponenty PxWebQuery
C – (Cancel) zrušenie zmien dát po editácii, alebo vkladaní nového riadku komponenty PxWebQuery

Tu v ďalšom sú popísané tlačítka, ktoré prislúchajú jednotlivým operáciam , a konkrétnemu písmenu, ktoré sa zapíše do property PxVisibleButtons, a to dané tlačítko sa zobrazí:

- Operácia First, do property PxVisibleButtons zapíšeme písmeno „F“

- Operácia Prior, do property PxVisibleButtons zapíšeme písmeno „R“

- Operácia Next, do property PxVisibleButtons zapíšeme písmeno „N“

- Operácia Last, do property PxVisibleButtons zapíšeme písmeno „L“

- Operácia Insert, do property PxVisibleButtons zapíšeme písmeno „I“

- Operácia Edit, do property PxVisibleButtons zapíšeme písmeno „E“

- Operácia Delete, do property PxVisibleButtons zapíšeme písmeno „D“

- Operácia Post, do property PxVisibleButtons zapíšeme písmeno „P“

- Operácia Cancel, do property PxVisibleButtons zapíšeme písmeno „C“

1.10. PxJSDatePicker – komponenta pre zadanie dátumu, postavená na JavaScripte

Komponenta PxJSDatePicker, je komponentou kalendárika založeného na JavaScripte. Táto komponenta slúži na výber dátumu v prehľadnom menu, výber aktuálneho alebo zvoleného dátumu, poprípade zvolenie režimu ručného zadania dátumu. Dole na obrázku je zobrazený kalendárik komponenty PxJSDatePicker:

Pokiaľ chceme pracovať s komponentou PxJSDatePicker, musíme si ju v súbore AddAdresar.aspx zadefinovať nasledovne:

```
<Prx:PxJSDatePicker ID="edtCREATE_DATE" runat="server" AddTableRow="True"></Prx:PxJSDatePicker>
```

V súbore AddAdresar.aspx.cs pripojíme komponentu PxJSDatePicker na komponentu PxWebQuery nasledovným spôsobom:

```
edtCREATE_DATE.PxDataSource = wquAdresar;
edtCREATE_DATE.FieldName = wquAdresar.Columns["CREATE_DATE"].ToString();
```

Na zarovnanie komponenty sa opäť používajú property AddTableRow, TableBegin a TableEnd, ktoré sú dobre popísané a rozobrané v časti kde je popísaná komponenta PxEdit.

1.11. PxDatePicker – komponenta pre zadanie dátumu

Komponenta PxDatePicker, je komponentou kalendárika. Táto komponenta slúži na výber dátumu v prehľadnom menu, výber aktuálneho alebo zvoleného dátumu, poprípade zvolenie režimu ručného zadania dátumu. Tento kalendárik, nevyužíva JavaScript, a pri rozbalený jeho menu, sa nám bude naťahovať celý formulár. Preto doporučujeme používať skôr komponentu PxJSDatePicker. Dole na obrázku je zobrazený kalendárik komponenty PxDatePicker:

```
<Prx:PxDatePicker ID="edtCREATE_DATE" runat="server" AddTableRow="True"  ></Prx:PxDatePicker>
```

V súbore AddAdresar.aspx.cs pripojíme komponentu PxDatePicker na komponentu PxWebQuery nasledovným spôsobom:

```
edtCREATE_DATE.PxDataSource = wquAdresar;
edtCREATE_DATE.FieldName = wquAdresar.Columns["CREATE_DATE"].ToString();
```

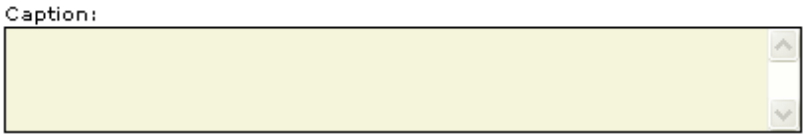
Na zarovnanie komponenty sa opäť používajú property AddTableRow, TableBegin a TableEnd, ktoré sú dobre popísané a rozobrané v časti kde je popísaná komponenta PxEdit.

1.12. PxMemo – komponenta, ktorá slúži na priamu editáciu dát, rozšírenie komponenty PxEdit

Komponenta PxMemo je obdobou komponenty PxEdit, a preto nemá veľký zmysel sa zaoberať popisom komponenty PxMemo, pokiaľ chcete vedieť viac o komponente PxMemo, vráťte sa ku kapitole, o komponente PxEdit. Komponenta PxMemo je obdobou komponenty PxEdit, a v časti OnInit, je pridaná navyše táto definícia TextBoxu, ktorý je súčasťou komponenty PxMemo:

```
Width = 400;
Height = 50;
TextMode = TextBoxMode.MultiLine;
```

To ako vyzerá komponenta PxMemo, je vidieť na obrázku dole:



Pokiaľ chceme pracovať s komponentou PxMemo, musíme si ju v súbore AddAdresar.aspx zadefinovať nasledovne:

```
<Prx:PxMemo ID="memText" runat="server" AddTableRow="True" ></Prx:PxMemo>
```

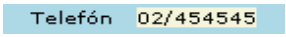
V súbore AddAdresar.aspx.cs pripojíme komponentu PxMemo na komponentu PxWebQuery nasledovným spôsobom:

```
memText.PxDataSource = wquAdresar;
memText.FieldName = wquAdresar.Columns["TEXT"].ToString();
```

Na zarovnanie komponenty sa opäť používajú property AddTableRow, TableBegin a TableEnd, ktoré sú dobre popísané a rozobrané v časti kde je popísaná komponenta PxEdit.

1.13. PxLabel – komponenta pre zobrazenie dát

Komponenta PxLabel slúži na zobrazenie dát prostredníctvom komponenty PxWebQuery. Dole na obrázku je jej zobrazenie:



Komponenta PxLabel sa skladá z dvoch labelov, viac schéma dole:

Label + Label

Prvý label slúži ako popisok(Caption), do druhého labelu sa zadáva hodnota(Value), respektíve sa pri prepojení automaticky načíta z komponenty PxWebQuery.

Pokiaľ chceme pracovať s komponentou PxLabel, musíme si ju v súbore InfoAdresar.aspx zadefinovať nasledovne:

```
<Prx:PxLabel ID="lblText" runat="server" AddTableRow="True"  ></Prx:PxLabel>
```

V súbore InfoAdresar.aspx.cs pripojíme komponentu PxLabel na komponentu PxWebQuery nasledovným spôsobom:

```
lblText.PxDataSource = wquAdresar;
lblText.FieldName = wquAdresar.Columns["TEXT"].ToString();
```

Na zarovnanie komponenty sa opäť používajú property AddTableRow, TableBegin a TableEnd, ktoré sú dobre popísané a rozobrané v časti kde je popísaná komponenta PxEdit.

Názov popiskov Caption, alebo názvy stĺpcov v gride(Title) môžete zadať centrálne, pri definícii komponenty PxWebQuery, kde hodnota v hranatých zátvorkách je názov stĺpca:

```
wquAdresar.Columns["Text"].Caption = "Text";
```

Zmena popisku komponenty PxLabel :

Zmenu popisku komponenty PxLabel, okrem centrálnej definície, môžeme vykonať nasledujúcim spôsobom:

```
lblText.Caption = "Text";
```

V ďalšom si uvedieme príklad z komponentou PxLabel. Vytvoríme si nový formulár z názvom „InfoAdresar“, a vzniknú nám dva súbory, a to „InfoAdresar.aspx“ a „InfoAdresar.aspx.cs“. Ešte pred tým než začneme programovať z novo vytvoreným formulárom, si v už vytvorenom fomuláry „Adresar.aspx“ doplníme property do komponenty PxSuperGrid, a to nasledovne:

```
PxInfoFormName="InfoAdresar.aspx"
```

Definícia komponenty PxSuperGrid, potom bude vyzerať nasledovne:

```
<Prx:PxSuperGrid ID="grdAdresar" runat="server"
  PageSize="4"
  PxEditFormName="AddAdresar.aspx"
  PxInfoFormName="InfoAdresar.aspx"
  PxVisibleButtons="SIDEO">
  <SelectedItemStyle BackColor="#ffddff" />
</Prx:PxSuperGrid>
```

Property PxInfoFormName, nám zabezpečí, že po kliknutí na tlačítko , ktoré tomu prislúcha v komponente PxSuperGridu, sa nám zavolá formulár „InfoAdresar.aspx“. Po stlačení tohto tlačítka sa nám tiež automaticky nastaví kurzor v komponente PxWebQuery. Potom nám stačí napojiť komponenty PxLabel vo formuláry „InfoAdresar.aspx“ a bude sa nám automaticky zobrazovať aktuálny riadok komponenty PxWebQuery, v komponentách PxLabel. Teda definícia celého formulára v súbore „InfoAdresar.aspx“, bude vyzerať nasledovne:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="InfoAdresar.aspx.cs" Inherits="InfoAdresar" %>

<%@ Register TagPrefix="Prx" Namespace="PxControls" Assembly="PxControls" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>Untitled Page</title>
</head>
<body>
   <form id="form1" runat="server">
   <div>
   <Prx:PxWebQuery ID="wquObec" runat="server" Value="wquObec"/>
   <Prx:PxWebQuery ID="wquAdresar" runat="server" Value="wquAdresar"/>

   <asp:Label ID="lblAddEditPanel" Font-Size="18pt" Font-Bold="true" runat="server" Text="Label"></asp:Label>
     <br /><br />

   <Prx:PxLabel ID="lblIDADRESAR" runat="server" AddTableRow="True" TableBegin="True" ></Prx:PxLabel>
   <Prx:PxLabel ID="lblName" runat="server" AddTableRow="True"  ></Prx:PxLabel>
   <Prx:PxLabel ID="lblPravnaForma" runat="server" AddTableRow="True"  ></Prx:PxLabel>
   <Prx:PxLabel ID="lblCREATE_DATE" runat="server" AddTableRow="True"  ></Prx:PxLabel>
   <Prx:PxLabel ID="lblInvalid_Adress" runat="server" AddTableRow="True" ></Prx:PxLabel>
   <Prx:PxLabel ID="lblADDRESS" runat="server" AddTableRow="True" TableEnd="True"></Prx:PxLabel>
   <Prx:PxFlyComboBox ID="fcmbMiesto" runat="server"></Prx:PxFlyComboBox>
   <br />
   <Prx:PxGreatRepeater ID="gtrTelefon" runat="server"></Prx:PxGreatRepeater>
   <br />
   <Prx:PxGreatRepeater ID="gtrFax" runat="server"></Prx:PxGreatRepeater>

   <br />
   <br />

   <asp:Button ID="btnClose" runat="server" Text="Zavrieť" onclick="ButtonClose_Click" />

   </div>
   </form>
</body>
</html>
```

V súbore „InfoAdresar.aspx.cs“ prepojíme komponenty PxLabel, s komponentou PxWebQuery nasledovne:

```
protected void Page_Load(object sender, EventArgs e)
 {
    lblAddEditPanel.Text = "Detail adresy";
```

```
        lblIDADRESAR.PxDataSource = wquAdresar;
        lblIDADRESAR.FieldName = wquAdresar.Columns["IDADRESAR"].ToString();

        lblName.PxDataSource = wquAdresar;
        lblName.FieldName = wquAdresar.Columns["NAME"].ToString();

        lblADDRESS.PxDataSource = wquAdresar;
        lblADDRESS.FieldName = wquAdresar.Columns["ADDRESS"].ToString();

        lblPravnaForma.PxDataSource = wquAdresar;
        lblPravnaForma.FieldName = wquAdresar.Columns["idpravnaforma"].ToString();

        lblCREATE_DATE.PxDataSource = wquAdresar;
        lblCREATE_DATE.FieldName = wquAdresar.Columns["CREATE_DATE"].ToString();

        lblInvalid_Adress.PxDataSource = wquAdresar;
        lblInvalid_Adress.FieldName = wquAdresar.Columns["Invalid_Adress"].ToString();

        fcmbMiesto.PxDataSource = wquAdresar;
        fcmbMiesto.FieldName = wquAdresar.Columns["miesto"].ColumnName;

        gtrTelefon.PxDataSource = wquAdresar;
        gtrTelefon.FieldName = "Telefon";

        gtrFax.PxDataSource = wquAdresar;
        gtrFax.FieldName = "Fax";
    }

    protected void ButtonClose_Click(object sender, EventArgs e)
    {
        this.Response.Redirect("~/Adresar.aspx");
    }
```

2. Inštalácia Px Frameworku

2.1 Inštalácia Px Frameworku na klientsky počítač

Vytvoríme si nový webový projekt v programe Visual Studio.
Rozbalíme súbor PxControls.zip. Po rozbalení tohto súboru, by sa nám mali objaviť adresáre „bin“ a adresár „images“.
Do adresára „bin“ nášho projektu, nakopírujeme súbory z adresára „PxControls\bin“.
Pokiaľ v našom projekte nemáme vytvorený adresár „bin“, tak ho vytvoríme,a nakopírujeme do neho súbory z adresára „PxControls\bin“.
Vytvoríme ďalší adresár, a to adresár z názvom „images“, a nakopírujeme do neho súbory z adresára „PxControls\images“.
Na všetky súbory *.dll v adresári „bin“ pridáme referenciu cez „Add Reference...“.

Ak chceme, aby komponenty Px Frameworku boli viditeľné v aspx súboroch, musíme si v hlavičké súboru zadefinovať „Prx“ prefix. Toto vykonáme nasledovne:

```
<%@ Register TagPrefix="Prx" Namespace="PxControls" Assembly="PxControls" %>
```

2.2. Inštalácia Px Frameworku na server

Pri inštalácii Px Frameworku, si budeme počínať obdobne, ako pri inštalácii na klientsky počítač.
Viac sa dočítate v časti „Inštalácia Px Frameworku na klientsky počítač.

3. Detailný popis programovania s komponentami Px Frameworku

3.2. PxWebQuery – komponenta pre prácu s databázou Oracle, detailný popis

3.1.1. Načítanie dát z databázy Oracle, pomocou komponenty PxWebQuery

Prostredníctvom komponenty *PxWebQuery,* môžeme celkom jednoducho načítať dáta z databázy a z takto načítanými dátami môžeme celkom ľahko pracovať. V našom príklade, do property ConnectString, priraďujeme ConnectionString,a do property SQLSelect, priraďujeme SQL príkaz, ktorý sa týka dát, ktoré chceme načítať. Potom načítanie dát vykonáme príkazom Open.

```
protected void Page_Load(object sender, EventArgs e)
{

if ((!IsPostBack)&&(wquAdresar.Active==false))
{
    wquAdresar.ConnectString = “"User Id=adr;Password=aa;Data Source=xe;";”;
    wquAdresar.SQLSelect = "select idadresar , name, address, create_date from Adresar";

    wquAdresar.Open();
}
}
```

3.1.2. Programové vkladanie nového riadku do databázy pomocou komponenty PxWebQuery

Na to aby ste prostredníctvom *PxWebQuery* vkladali nove riadky potrebujete vytvoriť sequenciu, z ktorej si komponenta *PxWebQuery* načíta jedinečné ID pre svoj primárny kľúč. Názov sequencie, by ste mali vytvoriť nasledujúcim spôsobom.

SEQ_“názov stĺpca primárneho kľúča“

Preto, keď tvoríte nové tabuľky nedávajte názov stĺpcu s primárnym kľúčom len „ID“, ale

napríklad ID+“názov tabuľky“.
V našom príklade, aby nám fungovalo aj vkladanie dát do databázy, vytvoríme sequenciu SQL príkazom:

```
CREATE SEQUENCE SEQ_IDADRESAR
  START WITH 1
  INCREMENT BY 1
  MAXVALUE 1E18
  NOMINVALUE
  NOORDER
  NOCACHE
  NOCYCLE;
```

Pozor !!! Samozrejme pod inými databázovými platformami, sa sequencia vytvára inak, viac sa môžete o tomto dočítať v štvrtej časti „ Špecifikácia nasadenia Px Frameworku pod rôznymi databázovými platformami (Oracle, MS SQL, MySQL, FireBird, InterBase)”.

Preto, keď zadáme príkaz wquAdresar.Insert(), komponenta PxWebQuery automaticky zavolá sequenciu, a tá vygeneruje nové ID pre primárny kľúč. My sa už nemusíme starať, o generovanie a priradenie jedinečného kľúča do fieldu, kde sa zadáva ID primárneho kľúča.
Viac príklad:

```
wquAdresar.Insert();
wquAdresar.Fields["name"] = "Grof Monte Christo";
wquAdresar.Fields["idpravnaforma"] = "Fyzická osoba";
wquAdresar.Fields["address"] = "Karlovarska 2";
wquAdresar.Fields["create_date"] = "04.07.2010";
wquAdresar.Fields["invalid_adress"] = "N";
wquAdresar.Post();
```

3.1.3. Programová editácia existujúceho riadku v databáze pomocou komponenty PxWebQuery

Editovať, opravovať už existujúci riadok, môžeme obdobne ako v predchádzajúcom príklade, len príkaz *Insert*, nahradíme príkazom *Edit*.

```
wquAdresar. Edit();
wquAdresar.Fields["name"] = "Juraj Peter Tarnoci";
wquAdresar.Fields["idpravnaforma"] = "Fyzická osoba";
wquAdresar.Fields["address"] = "Fatranska 22";
wquAdresar.Fields["create_date"] = "12.10.2010";
wquAdresar.Fields["invalid_adress"] = "N";
wquAdresar.Post();
```

Pokiaľ poznáme ID riadku, ktorý chceme editovať, môžeme si ho vyhľadať nasledujúcim spôsobom:

```
if (wquAdresar.FindByIdPK(6)!=-1)
    lblAdresar4.Text = wquAdresar.Fields["name"].ToString();
```

V našom prípade, číslo 6 je ID stĺpca primárneho kľúča riadku, ktorý hľadáme a do komponenty Label je zapísaný výsledok hľadania, a to hodnota stĺpca „name“.

3.1.4. Programové zmazanie existujúceho riadku v databáze pomocou komponenty PxWebQuery

Mazať už existujúci riadok, môžeme cez príkaz Delete, komponenty PxWebQuery. Pomocou príkazu FindByIdPK, môžeme vyhľadať riadok, ktorý chceme zmazať, a príkazom Delete vykonať mazanie, daného riadku.

```
if (wquAdresar.FindByIdPK(6)!=-1)
        wquAdresar.Delete();
```

3.1.5. Načítanie hodnôt z komponenty PxWebQuery cez cyklus while

Pokiaľ chceme načítať nejaké hodnoty, za všetky riadky v komponente PxWebQuery, môžeme to urobiť cez cyklus while. Tento spôsob práce s komponentou PxWebQuery , sa veľmi podobá na prácu s komponentami TTable v programovacom jazyku Delphi. Sú možné dva spôsoby načítania, prvý spôsob je načítanie od prvého riadku, ku poslednému. Druhý spôsob načítania, je od posledného riadku, ku prvému riadku.
Ďalej tu uvádzame výpis kódu, kde sú hodnoty načítané od prvého riadku ku poslednému a výsledok je potom priradený do komponenty Label.

```
wquAdresar.First();
string s=String.Empty;
while (!wquAdresar.Eof())
{
   s = s+", "+wquAdresar.Fields["name"].ToString();
   wquAdresar.Next();
}
lblAdresar.Text = s;
```

V ďalšom príklade je výpis kódu, kde sú hodnoty načítané od posledného riadku ku prvému a výsledok je potom priradený do komponenty Label.

```
wquAdresar.Last();
s = String.Empty;
while (!wquAdresar.Bof())
{
   s = s + ", " + wquAdresar.Fields["name"].ToString();
   wquAdresar.Prior();
}
lblAdresar2.Text = s;
```

Príkaz wquAdresar.Eof() vracia hodnotu true v prípade, keď sa príkazom Next(), už nemôžeme

posunúť na ďalší nasledujúci riadok. Podobne príkaz wquAdresar.Bof() vracia hodnotu true v prípade, keď sa príkazom Prior(), už nemôžeme posunúť na predchádzajúci riadok, pretože sa už nachádzame na prvom riadku.

3.1.6. Vyhľadávanie riadku v komponente PxWebQuery, podľa zadanej hodnoty a názvu stĺpca v ktorom sa má vyhľadávať

Pokiaľ potrebujem vyhľadávať v tabuľke komponenty PxWebQuery, môžem to vykonať cez príkaz *Find*, viac príklad:

```
if(wquAdresar.Find("%ško%","name")!=-1)
        lblAdresar3.Text = wquAdresar.Fields["name"].ToString();
```

Definícia príkazu Find:

```
public Int32 Find(string SearchValue,string sFieldName);
```

Popis parametrov príkazu Find:

Návratová hodnota príkazu Find je ItemIndex(index riadka, ktorý zodpovedá výberovým kritériám) komponenty PxWebQuery, v prípade že hľadaním sa nenájde žiaden riadok, potom návratová hodnota je -1.

SearchValue – hodnota, podľa ktorej sa vyhľadáva, v danom stĺpci zadanom v parametre sFieldName. Hodnotu SearchValue, môžeme vybaviť napr. percentami „%", funkcionalita je podobná, ako pri SQL príkaze „Like"
sFieldName – názov stĺpca, v ktorom je hľadaný daný reťazec

Ďalej komponenta PxWebQuery, umožňuje vyhľadávanie podľa primárneho kľúča, viac príklad dole:

```
if (wquAdresar.FindByIdPK(6)!=-1)
        lblAdresar4.Text = wquAdresar.Fields["name"].ToString();
```

Návratová hodnota príkazu FindByIdPK je ItemIndex(index riadka, ktorý zodpovedá výberovým kritériám) komponenty PxWebQuery, v prípade že hľadaním sa nenájde žiaden riadok, potom návratová hodnota je -1. Ako vyhľadávací parameter sa zadáva platná hodnota primárneho kľúča.

3.1.6. Nastavenie pozície kurzora riadku, v komponente PxSuperGrid

Nastavenie pozície kurzora v komponente PxSuperGrid môžeme priradeným hodnoty 0..RecordCount-1 do property ItemIndex komponenty PxSuperGrid.

```
if (wquAdresar.FindByIdPK(6) != -1)
   wquAdresar.ItemIndex = wquAdresar.FindByIdPK(6);
```

3.1.7. Výber, selekcia viacerých riadkov, ktoré vyhovujú výberovým kritériám

S pomocou komponenty PxWebQuery, môžeme urobiť aj výber viac riadkov ako je jeden, pokiaľ tak umožňujú výberové kritéria. Príkaz wquAdresar.DataSource() nám vracia objekt DataTable. Objekt DataTable podporuje selektovanie, výber riadkov. Viac príklad dole:

```
//Vypise cisla Id riadkov, ktore vyhovuju vyberovemu kriteriu
    DataRow[] dr = null;
    dr = wquAdresar.DataSource().Select("name like '%a%'", "name");

    ArrayList aList = new ArrayList();
    Int32 iOrdValue = 0;

    for (Int32 iCykl = 0; iCykl < dr.Length; iCykl++)
    {
       //Tato klauzula vykonava DISTINCT
       if (!aList.Contains(dr[iCykl].ItemArray.GetValue(iOrdValue).ToString()))
       {
          aList.Add(dr[iCykl].ItemArray.GetValue(iOrdValue).ToString());
       }
    }
    string s = String.Empty;
    for (Int32 iCykl2 = 0; iCykl2 < aList.Count; iCykl2++)
    {
       s = s + ", " + aList[iCykl2].ToString();
    }
    lblAdresar5.Text = s;
```

Táto rutinka nám urobí výpis všetkých id riadkov ktoré vyhovujú výberovému kritériu, v tomto prípade, sú to riadky u ktorých v stlpci „name“, sa v texte nachádza písmeno „a“.

3.1.8. Udalosti (Eventy) komponenty PxWebQuery

Komponenta PxWebQuery obsahuje aj udalosti(eventy), ktoré sú vyvolané pri zmene stavu komponenty. Každú udalosť komponenty PxWebQuery si môžeme definovať v procedúre „Page_Load“ formulára, v ktorom chceme vyvolať danú udalosť.
Definícia udalosti vyzerá nasledovne:

```
if ((!IsPostBack)&&(wquAdresar.Active==false))
{
   ….
   wquAdresar.AfterScroll += new PxControls.AfterScrollEventHandler(wquAdresar_AfterScroll);
   wquAdresar.Open();
    ….
}
else
{
   wquAdresar.AfterScroll += new PxControls.AfterScrollEventHandler(wquAdresar_AfterScroll);
}
```

Definícia procedúry „wquAdresar_AfterScroll“, ktorá je volaná udalosťou „AfterScroll“ bude vyzerať nasledovne:

```
void wquAdresar_AfterScroll(object sender, EventArgs e)
{
   lblAdresar.Text = wquAdresar.Fields["name"].ToString();
}
```

Pri každej zmene kurzora aktuálneho riadku, či už v komponente PxWebQuery, alebo v komponente PxSuperGrid, ktorá je napojená na komponentu PxWebQuery, je vyvolaná udalosť „AfterScroll“. Pri každom vyvolaný udalosti, je do komponenty Label zapísaná hodnota stĺpca „name“ aktuálneho riadku.

V ďalšom bude súhrnný popis všetkých udalostí, ktoré obsahuje komponenta PxWebQuery.

Názov udalosti	Popis udalosti
BeforeInsert	Táto udalosť je vyvolaná pred vykonaným príkazu PxWebQuery.Insert();
AfterInsert	Táto udalosť je vyvolaná po vykonaným príkazu PxWebQuery.Insert();
BeforeEdit	Táto udalosť je vyvolaná pred vykonaným príkazu PxWebQuery.Edit();
AfterEdit	Táto udalosť je vyvolaná po vykonaným príkazu PxWebQuery.Edit();
BeforePost	Táto udalosť je vyvolaná pred vykonaným príkazu PxWebQuery.Post();
AfterPost	Táto udalosť je vyvolaná po vykonaným príkazu PxWebQuery.Post();
BeforeValidPost	Táto udalosť je vyvolaná pri vykonaný verifikácie zadaných položiek, pred samotným uloženým dát metódou Post();
BeforeOpen	Táto udalosť je vyvolaná pred vykonaným príkazu PxWebQuery.Open();
AfterOpen	Táto udalosť je vyvolaná po vykonaný príkazu PxWebQuery.Open();
BeforeScroll	Táto udalosť je vyvolaná pred vykonaným zmeny pozície kurzora riadku v komponente PxWebQuery
AfterScroll	Táto udalosť je vyvolaná po vykonaným zmeny pozície kurzora riadku v komponente PxWebQuery
BeforeDelete	Táto udalosť je vyvolaná pred vykonaným príkazu PxWebQuery.Delete();
AfterDelete	Táto udalosť je vyvolaná po vykonaným príkazu PxWebQuery.Delete();
BeforeCancel	Táto udalosť je vyvolaná pred vykonaným príkazu PxWebQuery.Cancel();
AfterCancel	Táto udalosť je vyvolaná po vykonaným príkazu PxWebQuery.Cancel();
StateChange	Táto udalosť je vyvolaná pri zmene stavu komponenty PxWebQuery, a to obyčajne po vykonaný príkazov Insert, Edit, Delete, Post a pod.
AfterCreateField	Táto udalosť je vyvolaná po vytvorený všetkých stĺpcov(fieldov) tabuľky komponenty PxWebQuery. Táto udalosť sa využíva prevažne na vytvorenie nového dynamického stĺpca(fieldu) tabuľky komponenty PxWebQuery.

3.1.9. Vytvorenie nového dynamického stĺpca(field) v tabuľke komponenty PxWebQuery

Ak potrebujeme pri práci s komponentou PxWebQuery vytvoriť nový dynamický stĺpec môžeme tak urobiť prostredníctvom udalosti „AfterCreateField“.

```
if ((!IsPostBack)&&(wquAdresar.Active==false))
```

```
    {
      ….
      wquAdresar.AfterCreateField += new AfterCreateFieldEventHandler(AfterCreateField_Adresar);
      wquAdresar.Open();
       ….
    }
    else
    {
      wquAdresar.AfterCreateField += new AfterCreateFieldEventHandler(AfterCreateField_Adresar);
    }
```

V procedúre „AfterCreateField_Adresar“, si vytvoríme nový stĺpec a potom mu ešte priradíme popisok(Caption). Pri vytváraný stĺpca, musíme zadať taký názov stĺpca, ktorý ešte v tabuľke komponenty PxWebQuery neexistuje.

```
protected void AfterCreateField_SposOsoba(object sender, EventArgs e)
  {
      wquSposOsoba.DataSource().Columns.Add("FullName", Type.GetType("System.String"),
"ISNULL(Last_Name+' ','')" + "+ISNULL(First_Name+' ','')" + "+ISNULL(', '+Titl1,'')");
      wquSposOsoba.DataSource().Columns["FullName"].Caption="Priezvisko a Meno";


      wquSposOsoba.DataSource().Columns.Add("FullNameAll", Type.GetType("System.String"),
"ISNULL(Last_Name+' ','')" + "+ISNULL(First_Name+' ','')" + "+ISNULL(', '+Titl1,'')");
      wquSposOsoba.DataSource().Columns["FullNameAll"].Caption="Priezvisko, Meno, Titul";

  }
```

Pri tvorení expresion výrazu, pri vytváraní dynamického stĺpca môžeme požiť nasledujúce funkcie:

CONVERT

Popis	Prevedie výraz do presne špecifikovaného typu
Syntax	Convert(expression, type)
Argument	**expresion** – výraz, ktorý bude prevedený **type** - .NET Framework typ, na ktorý hodnota bude prevedená.

Príklad: myDataColumn.Expression="Convert(total, 'System.Int32')"

LEN

Popis	Vracia dĺžku reťazca
Syntax	LEN(expression)
Argument	**expression** -- reťazec, ktorý bude prevedený

Príklad: myDataColumn.Expression="Len(ItemName)"

ISNULL

Popis	Kontroluje sa výraz, pokiaľ nie je Null tak sa vráti, inak sa vráti náhradná hodnota
Syntax	ISNULL(expression, replacementvalue)
Argument	**expression** – výraz, ktorý sa kontroluje **replacementvalue** - Ak výraz vráti null, nahradí sa hodnotou z replacementvalue.

Príklad: myDataColumn.Expression="IsNull(price, -1)"

IIF

Popis	Vráti jednu z dvoch hodnôt, v závislosti od výsledku logického výrazu.
Syntax	IIF(expr, truepart, falsepart)
Argument	**expr** - výraz, ktorý sa vyhodnocuje **truepart** - vráti túto hodnotu, ak je výraz pravdivý. **falsepart** - vráti túto hodnotu, ak je výraz nepravdivý

Príklad: myDataColumn.Expression = "IIF(total>1000, 'expensive', 'dear')

TRIM

Popis	Odstráni všetky vedúce a koncové prázdne znaky, ako \ r \ n \ t "
Syntax	TRIM(expression)
Argument	**expression** – výraz, ktorý bude upravený

SUBSTRING

Popis	Vracia podreťazec určitej dĺžky, začínajúci na stanovenom miesto v reťazci
Syntax	SUBSTRING(expression, start, length)
Argumenty	**expression** - reťazec, z ktorého sa vytvára podreťazec **start** - celé číslo, ktoré určuje, kde podreťazec začína **length** - celé číslo, ktoré určuje dĺžku podreťazca

Príklad: myDataColumn.Expression = "SUBSTRING(phone, 7, 8)"

3.1.10. Procedúra ReOpen komponenty PxWebQuery a znovu načítanie dát do komponenty PxWebQuery

Ak chceme do komponenty PxWebQuery znova načítať dáta, takzvaný refresh, nemusíme to robiť cez zatváranie a znovu otváranie komponenty PxWebQuery, ale môžeme to urobiť cez procedúru „ReOpen()". Procedúra „ReOpen()" je veľmi rýchla, a doporučuje sa použiť pri zmene SQL príkazu. Viac príklad dole:

```
protected void Page_Load(object sender, EventArgs e)
{
 if ((!IsPostBack)&&(wquAdresar.Active==false))
 {
   wquAdresar.ConnectString = "User Id=adr;Password=aa;Data Source=xe;";
   wquAdresar.SQLSelect = "select idadresar , name, address, create_date from Adresar";

   wquAdresar.Open();
 }
 else
 {
 wquAdresar.SQLSelect = "select idadresar , name, address, create_date from Adresar
 where name like 'a%'";
 wquAdresar.ReOpen();
 }
 grdAdresar.PxDataSource = wquAdresar;
 grdAdresar.DataBind();
 }
```

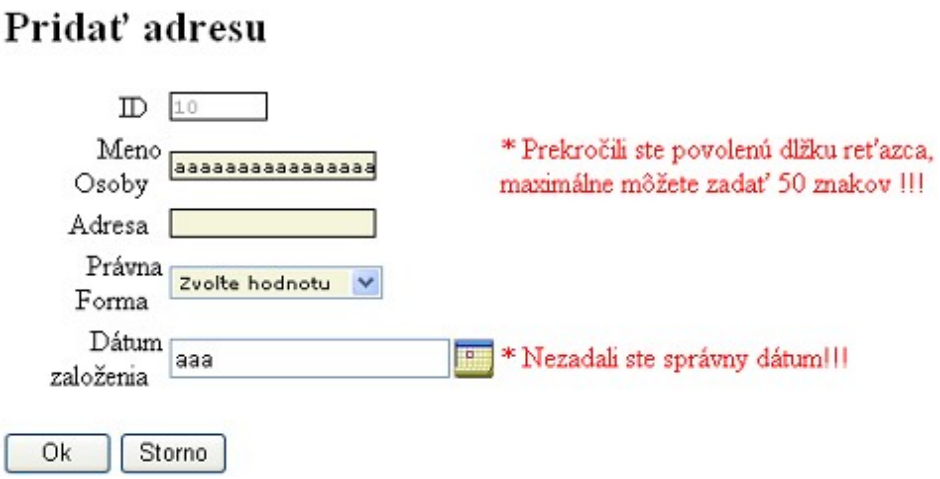
Iný prípad je, automatický refresh, ktorý môžeme spustiť na pozadí komponenty PxWebQuery. Komponenta PxWebQuery dokáže pri svojom behu, na pozadí vykonávať automatický refresh, a aktualizovať si obsah dát tabuľky. Tento refresh sa dá zapnúť nastavením property

AutomaticRefresh na hodnotu true. Do property IntervalQueryRefresh ešte môžeme zadať hodnotu intervalu v sekundách. Po vypršaní tohto intervalu, od posledného načítania alebo refrešovania dát, bude vykonaný nový refresh dát komponenty PxWebQuery. Pokiaľ do property IntervalQueryRefresh nezadáte žiadnu hodnotu, použije sa defaultna hodnota 120 sekúnd, teda ku automatickému refrešu bude dochádzať každé dve minúty.

```
wquAdresar.Open();
wquAdresar.AutomaticRefresh = true;
wquAdresar.IntervalQueryRefresh = 180; //v sekundach
```

3.1.11. Validácia, kontrola zadaných hodnôt prostredníctvom komponenty PxWebQuery a ostatných vizuálnych komponent(PxEdit, PxComboBox a atď.)

Komponenta PxWebQuery prostredníctvom komponenty PxEdit(ale aj PxComboBox, PxFlyComboBox, PxJSDatePicker a pod.) obsahuje automatickú validáciu. Ak stĺpec tabuľky, ktorý je pripojený na komponentu PxEdit je typu **string,** je automaticky prevádzaná validácia dĺžky reťazca, ak je typu **int**, validuje sa či bolo zadané celé číslo, ak je typu **date**, validuje sa či bol zadaný validný dátum a pod.
Na obrázku je príklad, kde sme zadali reťazec, ktorý svojou dĺžkou presahoval dĺžku reťazca stanoveného v databáze. Ďalej sme zadali dátum, ktorý nebol validný. Prepojením komponenty PxWebQuery a PxEdit sa vytvára automatická validácia. Ak zadaná hodnota v komponente PxEdit nekorešponduje z daným typom, neprejde cez automatickú validáciu, je automaticky zastavené ukladanie dát, a za komponentu PxEdit, je vypísaná chybová správa, prečo nedošlo ku uloženiu dát.

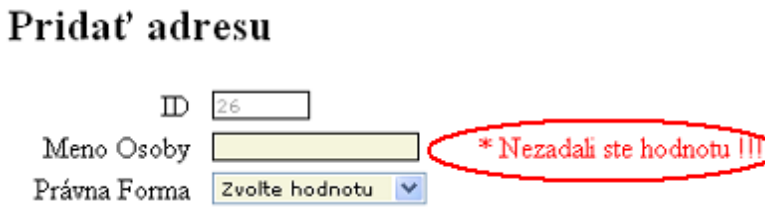


Okrem automatickej validácie, môžeme ešte definovať na komponente *PxWebQuery*, ďalšie doplnkové validácie, ktoré sa definujú cez parameter *AddParamValidation*.

```
wquAdresar.AddParamValidation("name", "vtIsNotNull");
```

V tomto prípade, pokiaľ nebude vyplnená položka „Meno Osoby“, tak nedôjde ku uloženiu dát, a pri stlačení tlačítka „Ok“, zostaneme v pôvodnom formulári, a za komponentu „Meno Osoby“ nám

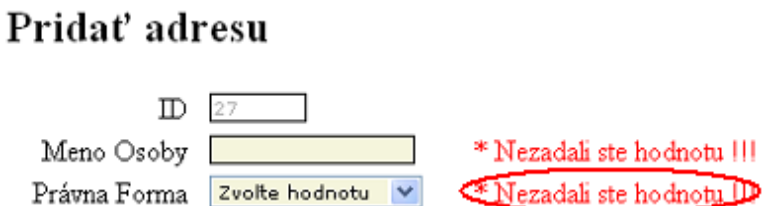
vypíše chybovú hlášku. Viac na obrázku dole:



Podobne ak na stĺpec právnej formy, tiež uvalíme parameter validácie s typom validácie „vtIsNotNull“, pri stlačení tlačítka „Ok“, zostaneme v pôvodnom formuláry pokiaľ, nevybereme nejakú hodnotu zo zoznamu „Právnej formy“. Príkaz na zadanie parametra validácie pre právnu formu bude vyzerať nasledovne:

```
wquAdresar.AddParamValidation("idpravnaforma", "vtIsNotNull");
```

Ak z comboboxu nevybereme žiadnu právnu formu, tak pri ukladaní dát, cez tlačítko „Ok“, ostaneme v pôvodnom formulári, a červenými písmenami bude zobrazená príčina prečo nedošlo ku uloženiu formulára, viac obrázok dole.



Definícia príkazu AddParamValidation:

```
public void AddParamValidation(string sFieldName, string sEnumValidationTest, string
sRegularExpression, Int32 iMinMaxValue, string sErrorMsg)

public void AddParamValidation(string sFieldName, string sEnumValidationTest, Int32
iMinMaxValue)

public void AddParamValidation(string sFieldName, string sEnumValidationTest)
```

Popis parametrov príkazu AddParamValidation:

sFieldName – názov, meno fieldu, stĺpca na ktorý sa má uvaliť daný typ validácie
sEnumValidationTest – typ validácie, kontroly, ktorá sa má vykonať, popis jednotlivých typov, bude uvedený ďalej
sRegularExpression – výraz pre zadanie validácie cez regulárne výrazy
iMinMaxValue – minimálna, maximálna hodnota výrazu, pri zadanom type fieldu typu integer
sErrorMsg – cez tento výraz, môžeme pri danom type validácie prepísať, alebo zadať novú chybovú správu

Popis jednotlivých typov validácie:

vtValidToDateTime – pri tomto type je testované, či zadaný výraz je platný dátum
vtIsNotNull – pri tomto type je testované, či bola zadaná nejaká hodnota, ak nie, bude vrátená chybová správa "Nezadali ste hodnotu !!!"
vtValidToEmail – pri tomto type je testované, či bola zadaná valídna, správna emailova adresa
vtValidToMaxNumber – pri tomto type je testované, či bolo zadané menšie číslo ako bolo definované v premennej iMinMaxValue, ak nie vráti chybovú správu
vtValidToMinNumber – pri tomto type je testované, či bolo zadané väčšie číslo ako bolo definované v premennej iMinMaxValue, ak nie vráti chybovú správu
vtValidToPSC – pri tomto type je testované, či bolo zadané správne poštové smerové číslo, ak nie vráti chybovú správu
vtValidToRegularExpression – pri tomto type je testované, či zadaná hodnota spĺňa zadaný regulárny výraz, ak nie vráti chybovú správu
vtValidToString – pri tomto type je testované, či bolo zadaný čisto len reťazec zložený z písmen, ak nie vráti chybovú správu
vtValidToTelefon – pri tomto type je testované, či bolo zadané validné telefónne číslo, ak nie vráti chybovú správu
vtValidToURL – pri tomto type je testované, či bola zadaná valídna hodnota URL, ak nie vráti chybovú správu
vtValidToCurrency – pri tomto type je testované, či bolo zadané validné číslo typu currency, ak nie vráti chybovú správu
vtValidToNumber – pri tomto type je testované, či bolo zadané validné, správne celé číslo, ak nie vráti chybovú správu
vtValidToFloat – pri tomto type je testované, či bolo zadané validné, správne reálne číslo, ak nie vráti chybovú správu

Každý typ validácie, má pevne pridelenú chybovú spravu. Ak by sme chceli túto chybovú spravu zmeniť, ak nám nevyhovuje jej doslovné znenie, môžeme to urobiť nasledujúcim spôsobom:

```
wquAdresar.AddParamValidation("name", "vtIsNotNull", "", 0, "Nezadali ste meno osoby !!!");
```

Ďalej si uvedieme príklad validácie cez regulárne výrazy.
Nie je obsahom, tejto knihy popisovať prácu z regulárnymi výrazmi, a to ako sa zostavujú, tu Vás môžeme odkázať len na početné zdroje na internete.

Tabuľku adresár sme rozšírili o ďalší stĺpec, a to o stĺpec na zadávanie PSČ.
Do súboru Adresar.aspx.cs, si validáciu prostredníctvom regulárneho výrazu zadefinujeme nasledovne:

```
string sRegVyrazToPSC = @"^\d{5}$|^\d{3}\s\d{2}$";
wquAdresar.AddParamValidation("psc", "vtValidToRegularExpression", sRegVyrazToPSC, 0, "Nezadali ste
platné poštové smerovacie číslo !!!");
```

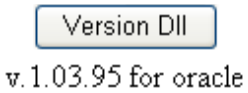
Zadefinovaný regulárny výraz nám umožní zadať 5 miestne číslo, a to buď z medzerou po 3 čísle, alebo bez medzery. Vždy keď chceme validovať cez regulárny výraz, musíme zadať typ validácie „vtValidToRegularExpresion". Ak sme si takto zadefinovali validáciu pre položku PSČ, tak pri nesprávnom zadaní PSČ, sa formulár neuloží, a zastaví sa jeho ukladanie. Za komponentou PSČ nám vráti chybovú správu, ktorú sme si zadefinovali vo výraze „ErrorMsg", viac obrázok.

3.1.12. Zistenie aktuálnej verzie Px Frameworku

Zistenie aktuálnej verzie Px Frameworku, môžete vykonať spustením nasledujúceho príkazu na komponente PxWebQuery:

```
protected void Button12_Click(object sender, EventArgs e)
{
  lblAdresar.Text = wquAdresar.Version;
}
```

Do komponenty Label (lblAdresar), bude priradený reťazec aktuálnej verzie Px Frameworku, a bude to vyzerať nasledovne:



Podľa výpisu môžeme identifikovať, že sa jedná o Px Framework, ktorý beží pod databázou Oracle, a je verzie 1.03.95.

3.1.13. Nastavenie jazykovej mutácie Px Frameworku

Px Framework momentálne podporuje nasledujúce jazykové mutácie, ich výpis nájdete dole.

Culture Name	Culture Identifier	Language-Country/Region
sk-SK	0x041B	Slovak - Slovakia
cs-CZ	0x0405	Czech - Czech Republic

en-US	0x0409	English - United States

Meniť jazykovú mutáciu, môžete zadaním reťazca „Culture Name“ do property CurrentCulture statického objektu PxCulture. Toto by sa malo vykonať hneď pri štarte aplikácie a v každom formulári, v procedúre „Page_Load“. Viac výpis zdrojového kódu dole:

```
protected void Page_Load(object sender, EventArgs e)
  {
    PxCulture.CurrentCulture = CultureInfo.CreateSpecificCulture("cs-CZ");
    ...
```

Pokiaľ by ste chceli zvoliť anglickú mutáciu, môžete to vykonať nasledujúcim spôsobom:

```
protected void Page_Load(object sender, EventArgs e)
  {
    PxCulture.CurrentCulture = new CultureInfo("en-US");
    ...
```

Zadanie jazykovej mutácie ovplyvňuje komponenty, ako sú PxSuperGrid, PxComboBox, PxFlyComboBox, PxJSDatePicker a objekty validácie, ktoré obsahujú texty chybových správ. Pokiaľ nezadáte jazykovú mutáciu, Px Framework si ju zistí sám z konfigurácie IIS servera, a tú bude používať defaultne pre všetky stránky.

3.2. Práca zo štýlmi, kaskádové štýly(CSS)

3.2.1. PxEdit – práca zo štýlmi, kaskádové štýly(CSS)

V tejto časti si popíšeme, ako pri komponente PxEdit pracovať zo štýlmi, ako deklarovať štýl pre všetky komponenty, z ktorých je zložená komponenta PxEdit. Komponenta PxEdit je zložená z týchto komponent:

Label + TextBox + Label

Preto každú z týchto komponent si v definícii CSS, si musíme deklarovať samostatne. Viac definícia kaskádového štýlu v súbore AddAdresar.aspx.

```
<style media="all" type="text/css">
 .textbox {
   // definícia štýlu pre TextBox
 }
 .textbox-label {
    // definícia štýlu pre Label
 }
 .textbox-rlabel {
   // definícia štýlu pre Rlabel, alebo aj inak Right Label
 }
</style>
```

Pri deklarácii štýlu komponenty PxEdit, si treba uvedomiť že komponenta TextBox je koreňová, a od nej sa dajú odvodiť názvy tried pre ostatne komponenty, z ktorých je zložená komponenta PxEdit. Ak chceme deklarovať štýl pre komponentu TextBox, ktorá je umiestnená v komponente PxEdit, môžeme tak urobiť cez property CssClass, do ktorej priradíme reťazec „textbox“. Na to

aby sme mohli deklarovať štýl aj pre popisky komponenty PxEdit (Caption), musíme pri definícii štýlov CSS zadefinovať triedy z nasledujúcimi názvami. Pre ľavý popisok zoberieme názov koreňovej triedy, v tomto prípade je to „textbox“ a spojíme ho z reťazcom, ktorý je presne definovaný, a to je „-label“. Takto sme sa dostali k názvu triedy, cez ktorú môžeme deklarovať štýl pre ľavý popisok(Caption), a ten názov je „textbox-label“. Podobne to je aj s pravým popiskom, tam výsledné meno triedy bude „textbox-rlabel“.

Názov komponenty, z ktorej je zložená komponenta PxEdit	Ako sa vytvára definícia názvu CssClasu pre jednotlivé komponenty obsiahnuté v komponente PxEdit pre CSS deklaráciu štýlov	Popis, o akú komponentu sa jedná
TextBox	CssClass	Editačné okienko TextBox
Label	CssClass+“-label“	Ľavý popisok (Caption)
RLabel	CssClass+“-rlabel“	Pravý popisok (Caption)

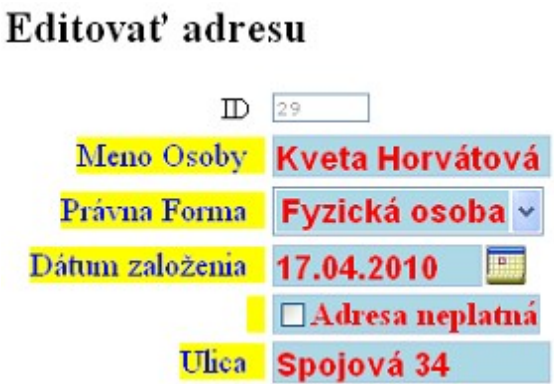
V súbore AddAdresar.aspx môžeme zadať štýl pre komponentu PxEdit nasledujúcim spôsobom:

```
<Prx:PxEdit ID="edtName" runat="server" CssClass="textbox" AddTableRow="True"  ></Prx:PxEdit>
```

V komponente PxEdit, sa pre komponenty Label a RLabel názov triedy pre CssClass modifikuje tak, že sa k textu obsiahnutému v property CssClass pridá ešte text „-label“, alebo „-rlabel“. Takže výsledná deklarácia všetkých štýlov pre všetky komponenty, v komponente PxEdit v CSS deklarácii bude vyzerať nasledovne:

```
<head runat="server">
   <title>Add Adresar - Px Framework</title>
   <style media="all" type="text/css">
         .textbox {
            background: lightblue;
            color: red;
            font-size: larger;
            height: 19px;
            width: 150px;
            font-weight: bold;
            border:solid 1px #b0c4de;
         }
         .textbox-label {
            background: yellow;
            color: Blue;
            font-size: larger;
            font-weight: bold;
            border: 0;
         }
         .textbox-rlabel {
            background: lightblue;
            color: Red;
            font-size: larger;
            font-weight: bold;
            border: 0;
         }
   </style>
</head>
```

Po deklarácii jednotlivých štýlov, ako to bolo deklarované hore vyššie, bude komponenta PxEdit vyzerať nasledovne:



3.2.2. PxJSDatePicker – práca zo štýlmi, kaskádové štýly(CSS)

V tejto časti si popíšeme, ako pri komponente PxJSDatePicker pracovať zo štýlmi, ako deklarovať štýl pre všetky komponenty, z ktorých je zložená komponenta PxJSDatePicker. Komponenta PxJSDatePicker je zložená z týchto komponent:

Label + TextBox + Image alebo Button +Label

Preto každú z týchto komponent si v definícii CSS, si musíme deklarovať samostatne. Viac definícia kaskádového štýlu v súbore AddAdresar.aspx.

```
<style media="all" type="text/css">
  .textbox {
    // definícia štýlu pre TextBox
  }
  .textbox-label {
     // definícia štýlu pre Label
  }
  .textbox-image {
     // definícia štýlu pre Label
  }
  .textbox-button {
     // definícia štýlu pre Label
  }
  .textbox-rlabel {
    // definícia štýlu pre Rlabel, alebo aj inak Right Label
  }
</style>
```

Pri deklarácii štýlu komponenty PxJSDatePicker, si treba uvedomiť že komponenta TextBox je koreňová, a od nej sa dajú odvodiť názvy tried pre ostatne komponenty, z ktorých je zložená

komponenta PxEdit. Ak chceme deklarovať štýl pre komponentu TextBox, ktorá je umiestnená v komponente PxEdit, môžeme tak urobiť cez property CssClass, do ktorej priradíme reťazec „textbox“. Na to aby sme mohli deklarovať štýl aj pre popisky komponenty PxJSDatePicker (Caption), musíme pri definícii štýlov CSS zadefinovať triedy z nasledujúcimi názvami. Pre ľavý popisok zoberieme názov koreňovej triedy, v tomto prípade je to „textbox“ a spojíme ho z reťazcom, ktorý je presne definovaný, a to je „-label“. Takto sme sa dostali k názvu triedy, cez ktorú môžeme deklarovať štýl pre ľavý popisok(Caption), a ten názov je „textbox-label“. Podobne to je aj s pravým popiskom, tam výsledné meno triedy bude „textbox-rlabel“.

Názov komponenty, z ktorej je zložená komponenta PxJSDatePicker	Ako sa vytvára definícia názvu CssClasu pre jednotlivé komponenty obsiahnuté v komponente PxJSDatePicker pre CSS deklaráciu štýlov	Popis, o akú komponentu sa jedná
TextBox	CssClass	Editačné okienko TextBox
Label	CssClass+“-label“	Ľavý popisok (Caption)
Image	CssClass+“-image“	Obrázkové tlačítko kalendárika
Button	CssClass+“-button“	Štandardné tlačítko kalendárika
RLabel	CssClass+“-rlabel“	Pravý popisok (Caption)

V súbore AddAdresar.aspx môžeme zadať štýl pre komponentu PxJSDatePicker nasledujúcim spôsobom:

```
<Prx:PxJSDatePicker ID="edtCREATE_DATE" runat="server" CssClass="textbox" AddTableRow="True"  >
</Prx:PxJSDatePicker>
```

V komponente PxJSDatePicker, sa pre komponenty Label, Image, Button a RLabel názov triedy pre CssClass modifikuje tak, že sa k textu obsiahnutému v property CssClass pridá ešte text „-label“, „-image“, „-button“alebo „-rlabel“.
Takže výsledná deklarácia všetkých štýlov pre všetky komponenty, v komponente PxJSDatePicker v CSS deklarácii bude vyzerať nasledovne:

```
<style media="all" type="text/css">
    .textbox {
      background: lightblue;
      color: red;
      font-size: larger;
      he-ight: 19px;
      width: 150px;
      font-weight: bold;

      border:solid 1px #b0c4de;
    }
    .textbox-label {
      background: yellow;
      color: Blue;
      font-size: larger;
      font-weight: bold;
      border: 0;
    }
    .textbox-rlabel {
      background: lightblue;
```

```
    color: Red;
    font-size: larger;
    font-weight: bold;
    border: 0;
  }
  .textbox-caption {
    background: lime;
    color: blue;
    font-size: larger;
    font-weight: bold;
    border: 0;
  }
  .textbox-button-insert
  {
     width:20px;
     heigth:17px;
  }
  .textbox-button-delete
  {
     width:20px;
     heigth:17px;
  }
</style>
```

3.2.3. PxGreatRepeater – práca zo štýlmi, kaskádové štýly(CSS)

V tejto časti si popíšeme, ako pri komponente PxGreatRepeater pracovať zo štýlmi, ako deklarovať štýl pre všetky komponenty, z ktorých je zložená komponenta PxGreatRepeater. Komponenta PxGreatRepeater môže byť zložená z týchto komponent:

PxEdit
PxCombobox
PxFlyCombobox
PxLabel

Každá z týchto komponent, je zložená z pod komponent, a preto pre každú z týchto podkomponent si v definícii CSS, si budeme deklarovať samostatnú triedu. Viac definícia kaskádového štýlu v súbore AddAdresar.aspx.

```
<style media="all" type="text/css">
  .textbox {
    // definícia štýlu pre TextBox
  }
  .textbox-label {
     // definícia štýlu pre Label
  }
  .textbox-image {
     // definícia štýlu pre Label
  }
  .textbox-button {
     // definícia štýlu pre Label
  }
  .textbox-rlabel {
```

```
                    // definícia štýlu pre Rlabel, alebo aj inak Right Label
                  }
                </style>
```

Okrem komponent z ktorých sa komponenta PxGreatRepeater dynamicky linkuje, sú tu komponenty, ktoré priamo súvisia s komponentou PxGreatRepeater, priamo ju vytvárajú. Sú to tlačítka na vytváranie a mazanie blokov dát, a popisok(Caption) komponenty PxGreatRepeater. Ku tlačítku „Insert“ alebo „+“ komponenty PxGreatRepeater sa vieme dostať cez triedu CssClass+“-button-insert“. Ku tlačítku „Delete“ alebo „-“ komponenty PxGreatRepeater sa vieme dostať cez triedu CssClass+“-button-delete“. Ku popisku (Caption) komponenty PxGreatRepeater sa vieme dostať cez triedu CssClass+“-caption“.

Potom ku priamosúvisiacim prvkom komponenty PxGreatRepeater, sa vieme dostať, cez nasledovnú definíciu CSS štýlu.

```
<style media="all" type="text/css">
    .textbox-button-insert {
      // definícia štýlu pre tlačítko „Insert“, alebo „+“
    }
    .textbox-button-delete {
      // definícia štýlu pre tlačítko „Delete“, alebo „-“
    }
    .textbox-caption {
      // definícia štýlu pre popisok(Caption) komponenty
        PxGreatRepeater
    }
  </style>
```

V súbore AddAdresar.aspx môžeme zadať štýl pre komponentu PxGreatRepeater nasledujúcim spôsobom:

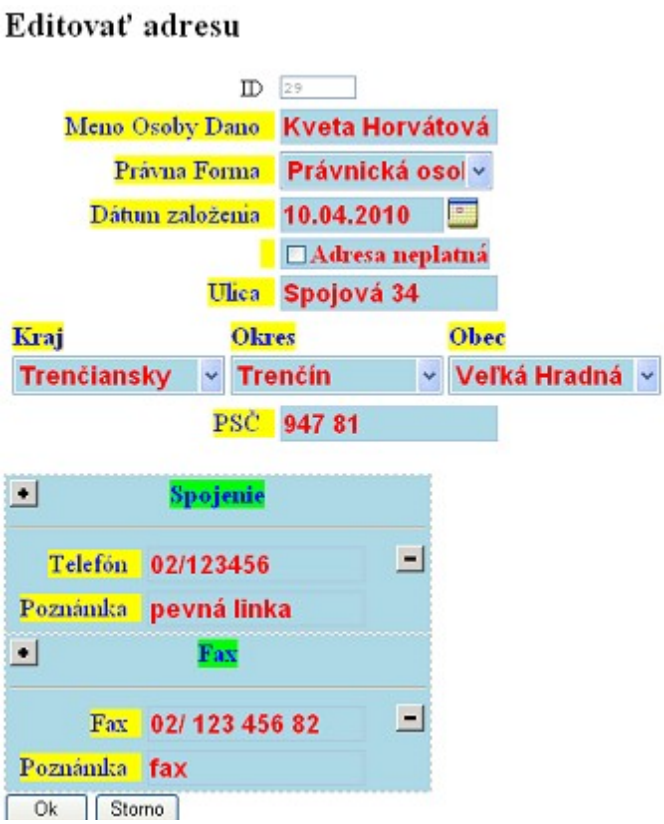
```
<br />
<Prx:PxGreatRepeater ID="gtrTelefon" runat="server" CssClass="textbox"></Prx:PxGreatRepeater>
<br />
```

Takže výsledná deklarácia všetkých štýlov pre všetky komponenty, v komponente PxGreatRepeater v CSS deklarácii bude vyzerať nasledovne:

```
<style media="all" type="text/css">
    .textbox {
      background: lightblue;
      color: red;
      font-size: larger;
      he-ight: 19px;
      width: 150px;
      font-weight: bold;

      border:solid 1px #b0c4de;
    }
    .textbox-label {
      background: yellow;
      color: Blue;
      font-size: larger;
      font-weight: bold;
      border: 0;
    }
```

```
.textbox-rlabel {
  background: lightblue;
  color: Red;
  font-size: larger;
  font-weight: bold;
  border: 0;
}
.textbox-caption {
  background: lime;
  color: blue;
  font-size: larger;
  font-weight: bold;
  border: 0;
}
.textbox-button-insert
{
   width:20px;
   heigth:17px;
}
.textbox-button-delete
{
   width:20px;
   heigth:17px;
}
</style>
```

Po aplikácii tohto štýlu, bude formulár AddAdresar.aspx vyzerať nasledovne:



Ostatné komponenty Px Frameworku, majú podobnú, alebo obdobnú prácu zo štýlmi, preto ju nebudeme popisovať ďalej.

4.1. Špecifikácia nasadenia Px Frameworku pod rôznymi databázovými platformami (Oracle, MS SQL, MySQL, FireBird, InterBase)

Pri nasadení pod rôznymi databázovými platformami sa Px Framework javí ako jednoliati celok, v skratke povedané, že aplikácia vytvorená pre databázu Oracle, sa bude dať jednoducho premigrovať napríklad pod databázu MS SQL. To čo bude potrebné upraviť, bude pozostávať z dvoch nasledujúcich veci, a to úpravou zdrojového kódu a štruktúry databázy.
Tieto úpravy sa budú týkať dvoch nasledujúcich vecí:

1. Zmena ConnectionStringu, ktorý sa zadáva do komponenty PxWebQuery pre danú databázovú platformu
2. Úprava databázy, ktorá sa bude týkať definície sequencií pre danú databázovú platformu

V ďalšom budú uvedené vzory ConnectionStringu, pre danú databázovú platformu, a tiež ako správne vytvárať sequencie (generátory), aby Px Framework s nimi správne pracoval. Sequencie(generátory) slúžia pri vytváraní nového riadku v tabuľke na definovanie jedinečného ID pre stĺpec primárneho kľúča.

4.1.1. Špecifikácia nasadenia Px Frameworku pod databázov Oracle

Na to, aby Px Framework pracoval správne s databázou Oracle, musíme do komponenty PxWebQuery, zadať správne reťazec ConnectionString. Dole je uvedený jeho vzor:

```
wquAdresar.ConnectString = "User Id=adr;Password=aa;Data Source=xe;";
```

Ďalej musíme pre každú tabuľku, do ktorej sa budú zadávať nové riadky vytvoriť sequenciu, ktorá slúži pri vytváraní nového riadku v tabuľke na definovanie jedinečného ID pre stĺpec primárneho kľúča.

Na to aby ste prostredníctvom *PxWebQuery* vkladali nove riadky, potrebujete vytvoriť sequenciu, z ktorej si komponenta *PxWebQuery* načíta jedinečné ID pre svoj primárny kľúč. Názov sequencie, by ste mali vytvoriť nasledujúcim spôsobom.

SEQ_“názov stĺpca primárneho kľúča“

Preto, keď tvoríte nové tabuľky nedávajte názov stĺpcu s primárnym kľúčom len „ID“, ale napríklad ID+“názov tabuľky“.
V našom príklade, aby nám fungovalo aj vkladanie dát do databázy, vytvoríme sequenciu SQL príkazom:

```
CREATE SEQUENCE SEQ_IDADRESAR
  START WITH 1
  INCREMENT BY 1
  MAXVALUE 1E18
  NOMINVALUE
  NOORDER
  NOCACHE
  NOCYCLE;
```

Px framework pre databázu Oracle, nepodporuje nasledujúce typy: Interval, Raw, Bfile, Long Raw, PLS_INTEGER, BINARY_INTEGER, XMLType.

4.1.2. Špecifikácia nasadenia Px Frameworku pod databázov MS SQL

Na to, aby Px Framework pracoval správne s databázou MS SQL, musíme do komponenty PxWebQuery, zadať správne reťazec ConnectionString. Dole je uvedený jeho vzor:

```
wquAdresar.ConnectString = "server=localhost\\SQLEXPRESS;Integrated Security=true;database=dbname;";
```

Ďalej musíme pre každú tabuľku, do ktorej sa budú zadávať nové riadky vytvoriť sequenciu, ktorá slúži pri vytváraní nového riadku v tabuľke na definovanie jedinečného ID pre každý riadok stĺpca primárneho kľúča.

Na to aby ste prostredníctvom *PxWebQuery* vkladali nové riadky, potrebujete vytvoriť sequenciu, z ktorej si komponenta *PxWebQuery* načíta jedinečné ID pre svoj primárny kľúč. MS SQL Server sequencie nepodporuje, preto bolo zvolené riešenie, ktoré tieto sequencie nahradzuje. Teda ak chcete pracovať s Px Frameworkom pod databázou MS SQL, musíte si vytvoriť sequenciu nasledujúcim spôsobom. Túto sequenciu si vytvoríte dvoma nasledujúcimi krokmi:

1. Vytvorenie tabuľky, kde sa odkladá posledná načítaná hodnota pre každú tabuľku
2. Vytvorenie procedúry, ktorá bude manažovať, načítanie a inkrementovanie hodnoty daného riadka tabuľky

Aby nám fungovalo aj vkladanie dát do databázy, vytvoríme SQL príkazom najprv tabuľku a potom procedúru. Názov tabuľky a procedúry musí zostať nezmenení.

Tabuľku pre generovanie sequencie, vytvoríte zadaním nasledujúceho SQL príkazu:

```
--VYTVORENIE SEQUENCII PRE TABULKU ADRESAR A PRAVNA FORMA
```

```
--VYTVORENIE TABULKY, KDE SA BUDU UKLADAT HODNOTY SEQUENCII
Create table All_Sequences (
    SeqName nvarchar(255) primary key, -- name of the sequence
    Start int not null default(1), -- seed value
    Increment int not null default(1), -- incremental
    CurrVal int
)
GO
```

Vytvorenie procedúry, ktorú využíva Px Framework na načítanie jedinečnej hodnoty kľúča pre danú tabuľku, vytvoríte zadaním nasledujúceho SQL príkazu:

```
--PROCEDURA NA NACITANIE SEQUENCIE
create procedure Get_Seguence
    @SeqName nvarchar(255),
    @result int OUTPUT,
    @start int = 1,
    @incr int = 1,
    @currvalue int = 1
as
begin
    declare @NewSeqVal int

    set NOCOUNT ON
    update All_Sequences
    set @NewSeqVal = CurrVal = CurrVal+Increment
    where SeqName = @SeqName

    if @@rowcount = 0 begin

      if exists (
         select 1 from All_Sequences
         where SeqName = @SeqName )
      begin
         print 'Sequence already exists.'
         return 1
      end

      if @start is null set @start = 1
      if @incr is null set @incr = 1
      set @currvalue = @start

      insert into All_Sequences (SeqName, Start, Increment, CurrVal)
      values (@SeqName, @start, @incr, @currvalue)

      set @result = @currvalue
      RETURN
    end
    set @result = @NewSeqVal
    return
end

GO
```

Hneď po vytvorení tabuľky a procedúry, môžeme začať pracovať s Px Frameworkom a vkladať nové riadky. Pokiaľ pre dané ID tabuľky, nebol v tabuľke All_Sequences vytvorení riadok, tak tento riadok sa automaticky vytvorí pri jej prvom zavolaní. Pri každom ďalšom zavolaní sa inkrementuje o hodnotu 1 smerom hore alebo o hodnotu Increment, zadanú v tabuľke All_Sequences.

Px framework pre databázu MS SQL, nepodporuje nasledujúce typy: Bit, SmallDateTime, Binary a VarBinary.

4.1.3. Špecifikácia nasadenia Px Frameworku pod databázov Mysql

Na to, aby Px Framework pracoval správne s databázou Mysql, musíme do komponenty PxWebQuery, zadať správne reťazec ConnectionString. Dole je uvedený jeho vzor:

```
wquAdresar.ConnectString = "Database=dbname;Data Source=localhost;User Id=root;Password=aaaa;";
```

Ďalej musíme pre každú tabuľku, do ktorej sa budú zadávať nové riadky vytvoriť sequenciu, ktorá slúži pri vytváraní nového riadku v tabuľke na definovanie jedinečného ID pre každý riadok stĺpca primárneho kľúča.

Na to aby ste prostredníctvom *PxWebQuery* vkladali nove riadky, potrebujete vytvoriť sequenciu, z ktorej si komponenta *PxWebQuery* načíta jedinečné ID pre svoj primárny kľúč. Mysql Server sequencie nepodporuje, preto bolo zvolené riešenie, ktoré tieto sequencie nahradzuje. Teda ak chcete pracovať s Px Frameworkom pod databázou Mysql, musíte si vytvoriť sequenciu nasledujúcim spôsobom. Túto sequenciu si vytvoríte dvoma nasledujúcimi krokmi:

1. Vytvorenie tabuľky, kde sa odkladá posledná načítaná hodnota pre každú tabuľku
2. Vytvorenie procedúry, ktorá bude manažovať, načítanie a inkrementovanie hodnoty daného riadka tabuľky

Aby nám fungovalo aj vkladanie dát do databázy, vytvoríme SQL príkazom najprv tabuľku a potom procedúru. Názov tabuľky a procedúry musí zostať nezmenení.

Tabuľku pre generovanie sequencie, vytvoríte zadaním nasledujúceho SQL príkazu:

```
--VYTVORENIE TABULKY PRE SEQUENCIU,
--GENEROVANIE NOVEHO ID PRE MYSQL
create table All_Sequence(
  seq_name varchar(30),
  start int,
  increment int,
  currval int,
  flag char(1));
```

Vytvorenie procedúry, ktorú využíva Px Framework na načítanie jedinečnej hodnoty kľúča pre danú tabuľku, vytvoríte zadaním nasledujúceho SQL príkazu:

```
--VYTVORENIE PROCEDURY SEQUENCIE
CREATE DEFINER = 'root'@'localhost'
PROCEDURE Get_Sequence(
      in SeqName varchar(255),
      out NextVal int
   )
   NOT DETERMINISTIC
   CONTAINS SQL
   SQL SECURITY DEFINER
   COMMENT ''
begin

declare rowcount int;
declare v_flag char(1);

select Count(*) into rowcount from All_Sequence where seq_name=SeqName;
if rowcount=0 then
  insert into All_Sequence values(SeqName,1,1,1,'A');
  set NextVal = 1;
else
  set v_flag='N';
  while (v_flag='N') do
```

```
    select flag into v_flag from All_Sequence where seq_name=SeqName;
   end while;

   update All_Sequence set flag='N' where seq_name=SeqName;
   select currval+INCREMENT into NextVal from All_Sequence where seq_name=SeqName;
   update All_Sequence set flag='A',currval=NextVal where seq_name=SeqName;
  end if;
 end;
```

Do sekcie „CREATE DEFINER“ procedúry, treba ešte zmeniť meno užívateľa a meno hostu na ktorom beží MySQL databáza, v našom prípade to je „'root'@'localhost'“.
Hore uvedená procedúra sa dala skompilovať bez chyby na servery MySQL verzia 5.1..
Pokiaľ, by nám neprešla kompilácia hore uvedenej procedúry, ďalej uvádzam upravenú procedúru, ktorá by mala zbehnúť na MySQL servery verzia 5.0.. a nižšej verzii, tu sú doplnené definície delimiterov.

```
DELIMITER $$

CREATE DEFINER = 'root'@'localhost'
PROCEDURE Get_Sequence(
      in SeqName varchar(255),
      out NextVal int
   )
   NOT DETERMINISTIC
   CONTAINS SQL
   SQL SECURITY DEFINER
   COMMENT ''
begin


declare rowcount int;
declare v_flag char(1);

select Count(*) into rowcount from All_Sequence where seq_name=SeqName;
if rowcount=0 then
  insert into All_Sequence values(SeqName,1,1,1,'A');
  set NextVal = 1;
else
  set v_flag='N';
  while (v_flag='N') do
    select flag into v_flag from All_Sequence where seq_name=SeqName;
  end while;

  update All_Sequence set flag='N' where seq_name=SeqName;
  select currval+INCREMENT into NextVal from All_Sequence where seq_name=SeqName;
  update All_Sequence set flag='A',currval=NextVal where seq_name=SeqName;
 end if;
end$$

DELIMITER ;
```

Hneď po vytvorení tabuľky a procedúry, môžeme začať pracovať s Px Frameworkom a vkladať nové riadky. Pokiaľ pre dané ID tabuľky, nebol v tabuľke All_Sequence vytvorení riadok, tak tento riadok sa automaticky vytvorí pri jej prvom zavolaní. Pri každom ďalšom zavolaní sa inkrementuje o hodnotu 1 smerom hore alebo o hodnotu Increment, zadanú v tabuľke All_Sequence.

Px framework pre databázu MySQL, nepodporuje nasledujúce typy: Bool, TimeStamp.

4.1.4. Špecifikácia nasadenia Px Frameworku pod databázov FireBird

Na to, aby Px Framework pracoval správne s databázou FireBird, musíme do komponenty PxWebQuery, zadať správne reťazec ConnectionString. Dole je uvedený jeho vzor:

```
wquAdresar.ConnectString = "DataSource = localhost;Database =D:\\Program
Files\\Firebird\\DATA27.FDB;UserID = SYSDBA;Password = masterkey;";
```

Ďalej musíme pre každú tabuľku, do ktorej sa budú zadávať nové riadky vytvoriť generátor, ktorý slúži pri vytváraní nového riadku v tabuľke na definovanie jedinečného ID pre stĺpec primárneho kľúča.

Na to aby ste prostredníctvom *PxWebQuery* vkladali nove riadky, potrebujete vytvoriť generátor, z ktorej si komponenta *PxWebQuery* načíta jedinečné ID pre svoj primárny kľúč. Názov generátora, by ste mali vytvoriť nasledujúcim spôsobom.

SEQ_“názov stĺpca primárneho kľúča“

Preto, keď tvoríte nové tabuľky nedávajte názov stĺpcu s primárnym kľúčom len „ID“, ale napríklad ID+“názov tabuľky“.
V našom príklade, aby nám fungovalo aj vkladanie dát do databázy, vytvoríme generátor SQL príkazom:

```
CREATE GENERATOR “SEQ_IDADRESAR“;
```

4.1.5. Špecifikácia nasadenia Px Frameworku pod databázov InterBase

Na to, aby Px Framework pracoval správne s databázou InterBase, musíme do komponenty PxWebQuery, zadať správne reťazec ConnectionString. Dole je uvedený jeho vzor:

```
wquAdresar.ConnectString = "DataSource = localhost;Database
=C:\\WINNT\\DATA27.GDB;UserID = SYSDBA;Password = masterkey;character
set=WIN1250;";
```

Pokiaľ pracujete aj s diakritikou, je potrebné do ConnectionStringu zadať aj character set, pretože provider pre interbase, dodávaný s Px Frameworkom, vyhadzuje chybovú hlášku pri zadaní diakritiky, a nezadefinovaní tejto znakovej sady do prihlasovacieho reťazca.

```
character set=WIN1250;
```

Ďalej musíme pre každú tabuľku, do ktorej sa budú zadávať nové riadky, vytvoriť generátor, ktorý slúži pri vytváraní nového riadku v tabuľke na definovanie jedinečného ID pre stĺpec primárneho kľúča.

Na to aby ste prostredníctvom *PxWebQuery* vkladali nove riadky, potrebujete vytvoriť generátor, z ktorej si komponenta *PxWebQuery* načíta jedinečné ID pre svoj primárny kľúč. Názov generátora,

by ste mali vytvoriť nasledujúcim spôsobom.

SEQ_“názov stĺpca primárneho kľúča“

Preto, keď tvoríte nové tabuľky nedávajte názov stĺpcu s primárnym kľúčom len „ID“, ale napríklad ID+“názov tabuľky“.
V našom príklade, aby nám fungovalo aj vkladanie dát do databázy, vytvoríme generátor SQL príkazom:

```
CREATE GENERATOR “SEQ_IDADRESAR“;
```

4.1.6. Špecifikácia nasadenia Px Frameworku všeobecne

Px Framework bol testovaný pod nasledujúcimi databázovými platformami:

Oracle server v.10.2.0.4.0.
Oracle server 10g Express Edition
MS SQL server 2005
MySQL server v.5.1.46. Win32
MySQL server v.5.0.51b
FireBird server v.2.0.5.
InterBase server WI-V6.5.0.28.

5.2. Pomocné komponenty Px Frameworku

5.1.1. Komponenta pre načítanie dát z databázy - SQLCoreQuery

Komponenta SQLCoreQuery slúži na vkladanie, editovanie, načítanie a mazanie dát v databáze. Nad komponentou SQLCoreQuery je vystavaná komponenta PxWebQuery. Komponenta SQLCoreQuery, nepodporuje žiadne ukladanie svojich štruktúr do objektov sessions alebo ViewState, tak ako je to u komponenty PxWebQuery. Viac príklad dole, ako sa dá načítať obsah tabuľky z databázy do gridu.

```
protected void Button5_Click(object sender, EventArgs e)
  {
     SQLCoreQuery quPravForm = new SQLCoreQuery();

     quPravForm.ConnectString = "User Id=skodapeter;Password=aa;Data Source=xe;";

     quPravForm.SQLText = "SELECT * FROM pravform";
     quPravForm.Open();

     StringBuilder sb = new StringBuilder();

     sb.Append("<TABLE border=1>");

     while (quPravForm.Read())
     {
        sb.Append("<TR>");
        for (int iCykl = 0; iCykl < quPravForm.FieldCount; iCykl++)
        {
           sb.Append("<TD>");
           sb.Append(quPravForm.Fields(iCykl).ToString());
        }
     }
     sb.Append("</TABLE>");

     LiteralControl tbl = new LiteralControl(sb.ToString());

     this.Page.Controls.Add(tbl);

    lblAdresar3.Text = sb.ToString();

    quPravForm.Close();
  }
```

Na to aby sme mohli pracovať s komponentou SQLCoreQuery, musime si do klauzuly uses zadefinovať assembly s nazvom „Component“.

```
using Component;
```

Záver

Filozofiou Px Frameworku je, že komponenta PxWebQuery je srdce aplikácie(Core), ktorá poskytuje, alebo v sebe obsahuje mohutnosť dát, definície prepojení tabuliek, definície zobrazení niektorých položiek v komponente PxSuperGrid, definície radenia dát v komponentách, ako je PxFlyCombobox, PxGreatRepeater apod. a plus ešte modul validácie dát pri ich ukladaní. Komponenta PxWebQuery požičiava dáta zo svojej mohutnosti, ostatným kompozitným komponentám, ako sú komponenty PxEdit, PxCombobox, PxCheckbox, PxFlyCombobox, PxGreatRepeater a pod., ktoré pracujú v presne vymedzenom diapazóne stanovenom komponentou PxWebQuery.

Px Framework, to čo do doby svojho trvania je veľmi mladý, a preto veľmi radi uvítame pripomienky a návrhy, ktoré by mohli prispieť ku skvalitneniu komponent, práce s nimi a ich všestranosti.

Všetkým priaznivcom Px Frameworku prajeme, veľa úspechov pri práci s Px Frameworkom, a vytrvalosť pri prenikaní do tajov práce a filozofie na základe ktorej bol Px Framework vytvorení. Veľa úspechov Vám praje realizačný team Px Frameworku.