

Px Framework v.1.70.01 for the Programings to Asp.Net application

Px Framework v.1.70 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 (relácií) 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 kategórií a podkategórií)
- 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. **PxJSPicker** – 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
- 1.14. **PxCheckBoxList** – komponenta pre zobrazenie a výber hodnôt zo zoznamu
- 1.15. **PxRadioButtonList** – komponenta pre zobrazenie a výber hodnoty zo zoznamu
- 1.16. **PxChart** – komponenta pre zobrazenie a prácu s grafmi
- 1.17. **PxFilterView** – vizuálna komponenta pre filtrovanie obsahu dát tabuľky v komponente PxWebQuery
- 1.18. **PxUploader** – komponenta na nahrávanie binárnych a textových súborov na server.
- 1.19. **PxDBPopUp** – komponenta pre načítanie dát z komponenty PxWebQuery a ich zobrazenie v prehľadnej tabuľke, výber hodnoty v tabuľke a jej následne zapísanie do editačného prvku, napr. PxEdit, TextBox, PxComboBox apod.
- 1.20. **PxLogin** – komponenta pre autorizáciu a prihlásenie sa do aplikácie
- 1.21. **PxSpeedButton** – varianta tlačítka s obrázkom
- 1.22. **PxMenu** – komponenta pre zobrazenie dynamického menu
- 1.23. **PxColorComboBox** – komponenta pre výber konkrétnej farby z ComboBoxu s farebným rozlíšením
- 1.24. **PxCounter** – komponenta pre zobrazenie počtu prístupov na stránku

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. Nastavenie pozície kurzora riadku, v komponente PxSuperGrid
- 3.1.8. Výber, selekcia viacerých riadkov, ktoré vyhovujú výberovým kritériám
- 3.1.9. Udalosti (Eventy) komponenty PxWebQuery
- 3.1.10. Vytvorenie nového dynamického stĺpca(field) v tabuľke komponenty PxWebQuery
- 3.1.11. Procedúra ReOpen komponenty PxWebQuery a znovu načítanie dát do komponenty PxWebQuery
- 3.1.12. Validácia, kontrola zadáných hodnôt prostredníctvom komponenty PxWebQuery a ostatných vizuálnych komponent(PxEdit, PxComboBox a atď.)
- 3.1.13. Zistenie verzie Px Frameworku
- 3.1.14. Nastavenie jazykovej mutácie Px Frameworku
- 3.1.15. Ďalšie metódy AddParam... komponenty PxWebQuery
 - 3.1.15.A. Metóda AddParamLink komponenty PxWebQuery
 - 3.1.15.B. Metóda AddParamDynamicField komponenty PxWebQuery
- 3.1.16. Špeciálna funkcia GetValueFromKey komponenty PxWebQuery
- 3.1.17. Tri spôsoby použitia komponenty PxWebQuery
- 3.1.18. Vytvorenie novej jazykovej mutácie Px Frameworku

3.2. Práca so štýlmi, kaskádové štýly(CSS)

- 3.2.1. PxEdit – práca so štýlmi, kaskádové štýly(CSS)
- 3.2.2. PxJSDatePicker – práca so štýlmi, kaskádové štýly(CSS)
- 3.2.3. PxGreatRepeater – práca so 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, natiehanie dát z databázy, ich zobrazenie a úpravu (vkládanie 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 zadať prefix „Prx“. 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.ConnectionString = "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.ConnectionString = "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="SIDE" >  
</Prx:PxSuperGrid>
```

Komponenta *PxWebQuery* automaticky poskytuje pri prepojení s *PxSuperGridom* funkcie, ako vkladanie dát, editovanie, mazanie, sortovanie a stránkovanie(paging).

Na to aby ste prostredníctvom *PxWebQuery* vkladali nové riadky, potrebujete vytvoriť sekvenciu, z ktorej si komponenta *PxWebQuery* načíta jedinečné ID pre svoj primárny kľúč. Názov sekvencie, 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 sekvenciu 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 sekvencia 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

◀ ◁ ▷ ▶

Do komponenty *PxWebQuery* bola pridaná zaujímavá property "Cached". Táto property slúži na kešovanie dát medzi užívateľmi v stave, keď sa obsah dát tabuľky nemení a query je využívaná len na čítanie dát. Toto zrýchľuje načítanie dát a znižuje počet prístupov do databázy.

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 informácií 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.ConnectionString = "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);
```

```
public void AddParamKey(string aFieldName, string aFieldNameKey, string aFieldNameValue,
    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 vhodné 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 funkcionality komponenty *PxSuperGrid* pri zadaní 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.

...	IDADRESAR	NAME	ADDRESS	IDPRAVNAFORMA
	1	Peter ŠKODA	Banská Bystrica	Fyzická osoba
✓✗	2	Ján Kolenička	Nitra	Fyzická osoba
	3	Palma Duffy	Bratislava	Fyzická osoba
	4	Jan Shrek	Dolné Kosiby	Právnická osoba

Page 1 [2]

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)
```

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

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

```
public void AddParamWebQuery(string aFieldName, string aFieldNameKey, string aFieldNameValue,
    PxWebQuery aPxWebQuery)
```

Popis parametrov:

aFieldName – Meno stĺpca v tabuľke na ktorý sa aplikuje metóda *AddParamWebQuery*

aFieldNameKey – Meno stĺpca číselníka, ktorý reprezentuje kľúč 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 vhodné označiť ako Key a Value, kvôli lepšej prehľadnosti

aPxWebQuery – Komponenta *PxWebQuery* v ktorej je cez SQL príkaz načítaný číselník, jednotlivé stĺpce je vhodné označiť ako Key a Value, kvôli lepšej prehľadnosti

Príklad zadania v súbore Adresar.aspx.cs v metóde Page_Load:

```
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

...	ID	Názov	Názov (Name)
 	2	Fyzická osoba	FO
 	1	Právnická osoba	PO
 PRIDAŤ NOVÝ RIADOK...			
Page 1			
   			

Takto vyzerá editácia prostredníctvom komponenty PxSuperGrid.

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

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.ConnectionString = "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="SIDE0">
</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 *PxSuperGríde*(ich zobrazenie alebo skrytie), použijeme property *PxVisibleButtons* komponenty *PxSuperGridu*.

```
grdAdresar.PxVisibleButtons="SIDE0";
```

Tu v ďalšom je popísaná legenda, podľa ktorej sa zobrazujú jednotlivé tlačítka, pokiaľ je tam zapísané nejaké písmeno, tak sa v PxSuperGríde, 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*, stačí keď, do property *PxEditFormName* zadáme cestu, alebo názov formulára kde bude táto editácia vykonaná.

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

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

```
<Prx:PxSuperGrid ID="grdAdresar" runat="server"  
    PageSize="4"  
    PxEditFormName="AddAdresar.aspx"  
    PxInfoFormName="InfoAdresar.aspx"  
    PxVisibleButtons="SIDE0">  
</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 popísané 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 zadať 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 vyselektovaný, vybraný riadok,

zabezpečme to zadáním nasledujúceho riadku do komponenty PxSuperGrid:

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

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

```
<Prx:PxSuperGrid ID="grdAdresar" runat="server"
    PageSize="4"
    PxEditFormName="AddAdresar.aspx"
    PxInfoFormName="InfoAdresar.aspx"
    PxVisibleButtons="SIDE0">
    <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 zobrazení 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 zadať 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();
```

Pokiaľ nechceme editovať dáta v komponente PxSuperGrid, môžeme to zabezpečiť nastavením property ReadOnly na True. Viac príklad dole:

```
grdAdresar.ReadOnly = true;
```

Komponenta PxSuperGrid podporuje aj zobrazenie počtu riadkov v dolnom pravom rohu. Toto zobrazenie môžeme aktivovať cez property *VisibleRecordCount*, nastavením na hodnotu true. Meniť

text môžeme cez property *CaptionRecordCount*.

```
grdAdresar.VisibleRecordCount = true;  
grdAdresar.CaptionRecordCount = "Počet záznamov:";
```

Zobrazenie počtu riadkov v komponente PxSuperGrid, môžete vidieť na dole uvedenom obrázku:

...	ID	Názov	Názov(Name)
 	2	Fyzická osoba	FO
 	1	Právnická osoba	PO
 	3	Juridical Person	JP
 PRIDAŤ NOVÝ RIADOK...			
Page 1			Počet záznamov: 3

Ak by sme chceli zadať popisok stĺpca(column) komponenty PxSuperGrid, tzv. ToolTip, môžeme to urobiť cez property *Header.ToolTip*. Ak by sme chceli, zmeniť názov stĺpca, môžeme to urobiť cez property *Header.Text*. Ak by sme chceli vypnúť sortovanie pre daný stĺpec, môžeme použiť property *Header.NoSort* a nastaviť ju na hodnotu *true*. Podobne, cez property *Header.ReadOnly*, môžeme vypnúť editáciu stĺpca v komponente PxSuperGrid. Priradenie týchto vlastností musíme urobiť pred priradením komponenty PxWebQuery do property *PxDataSource*.

```
grdAdresar.Header["name"].ToolTip = "Zadajte právnu formu !!!";  
grdAdresar.Header["name"].NoSort = true;  
grdAdresar.Header["name"].Text = "Právna Forma";  
grdAdresar.Header["ID"].ReadOnly = true;  
grdTyp_SposOsoba.PxDataSource = wquTyp_SposOsoba;
```

Na obrázku je zobrazený príklad zadania, hore uvedených vlastností(property).

...	ID	Právna Forma	Názov(Name)
 	2	Fyzická osoba	FO
	1	Právnická osoba	PO
	3	Juridical Person	JP
Page 1			Počet záznamov: 3

Komponenta PxSuperGrid obsahuje property *DateFormat*, pomocou ktorej môžeme ovládať zobrazenie dátumu v stĺpcoch typu *DateTime* komponenty PxSuperGrid. Pokiaľ chceme, aby sa nám v gride zobrazoval vedľa dátumu aj čas, použijeme masku „d.M.yyyy H:mm:ss“, ktorú zadáme do property *DataFormat*.

```
grdAdresar.DateFormat = "d.M.yyyy H:mm:ss";
```

Pokiaľ chceme zobrazovať čisto len dátum, použijeme masku „dd.MM.yyyy“. Pokiaľ by sme chceli tento dátum aj v gride editovať, použijeme radšej masku „d.M.yyyy“, lebo automatická validácia by nám mohla spôsobiť zastavenie ukladania dát v prípade keď by sme nezadali korektný dátum aj z

nulami, viac príklad zadania dátumu dole.

Príklad zadania masky a potom vstupnej hodnoty, aby došlo ku korektnému uloženiu hodnoty pri ukladaní a nebol zastavený proces ukladania automatickou validáciou.

```
"d.M.yyyy h:mm:ss"    -> 1.1.2011 3:26:13
"dd.MM.yyyy H:mm:ss"  -> 01.01.2011 15:26:13
"d.M.yyyy hh:mm:ss"   -> 1.1.2011 03:26:13
"dd.MM.yyyy HH:mm:ss" -> 01.01.2011 15:26:13
"dd.MM.yyyy"          -> 1.1.2011
```

Komponenta PxSuperGrid(ale aj PxEEdit, PxComboBox, PxFlyComboBox, PxJSDatePicker a pod.) obsahuje a podporuje automatickú a doplnkovú validáciu. Ak stĺpec tabuľky, ktorý je pripojený na komponentu PxSuperGrid 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 zadán validný dátum a pod. Viac o automatickej a doplnkovej 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 správny.



Keď nabehneme kurzorom myši nad ikonku s výkričníkom, bublinová nápoveda nám vypíše chybovú hlášku chyby, ktorá spôsobila zastavenie ukladania. V tomto prípade nebol zadán správny dátum.

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

Pomocou komponenty PxEEdit 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:PxDit ID="edtIDADRESAR" runat="server" AddTableRow="True" TableBegin="True" >
</Prx:PxDit>
<Prx:PxDit ID="edtName" runat="server" AddTableRow="True" ></Prx:PxDit>
<Prx:PxDit ID="edtADDRESS" runat="server" AddTableRow="True" ></Prx:PxDit>
<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 *PxDit* 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 *PxDit* 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 *PxDit*, bude popísaný dôvod prečo nedošlo ku automatickému uloženiu. Komponenta *PxDit* 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.sqlInsert)
        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:

Editovať adresu

ID	2
Meno Osoby	Ján Kolenička
Adresa	Nitra
Právna Forma	Fyzická osoba
Dátum založenia	11.02.2010
Ok Storno	

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="SIDE0">
    <SelectedItemStyle BackColor="#ffddff" />
</Prx:PxSuperGrid>
```

Po stlačení tlačítka „editovať“  v komponente PxSuperGridu , sa nám otvorí formulár AddAdresar.aspx.

Tam môžeme zadávať nové hodnoty, popřípade meniť staré hodnoty, po stlačení tlačítka „Ok“, sú tieto hodnoty uložené.

Komponenta PxEdit(ale aj PxSuperGrid, PxComboBox, PxFlyComboBox, PxJSDatePicker a pod.) obsahuje automatickú a doplnkovú 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 zadáný 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 správny.

Pridat' adresu

ID	10	
Meno	aaaaaaaaaaaaaaaa	* Prekročili ste povolenú dĺžku reťazca, maximálne môžete zadať 50 znakov !!!
Osoby		
Adresa		
Právna	Zvoľte hodnotu	
Forma	Zvoľte hodnotu	
Dátum založenia	aaa	* Nezadali ste správny dátum!!!

Polohovanie a zarovnanie prvkov PxEEdit:

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

Editovať adresu

ID	2	Meno Osoby
	Ján Kolenička	Adresa
	Nitra	Právna Forma
	Fyzická osoba	Dátum založenia
	11.02.2010	

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

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

Každá komponenta PxEEdit 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:PxEEdit ID="edtIDADRESAR" runat="server" AddTableRow="True" TableBegin="True" ></Prx:PxEEdit>
```

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

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

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

```
<Prx:PxEEdit ID="edtIDADRESAR" runat="server" AddTableRow="True" TableBegin="True" ></Prx:PxEEdit>
<Prx:PxEEdit ID="edtName" runat="server" AddTableRow="True" ></Prx:PxEEdit>
<Prx:PxEEdit ID="edtADDRESS" runat="server" AddTableRow="True" ></Prx:PxEEdit>
<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:

Editovať adresu

ID	2
Meno Osoby	Ján Kolenička
Adresa	Nitra
Právna Forma	Fyzická osoba
Dátum založenia	11.02.2010
Ok Storno	

Komponenta PxEEdit má ešte ďalšie property na zarovnanie svojho obsahu. Property AddTdToLabel, AddOnlyTdToLabel nám slúžia na zarovnanie prvkov komponenty PxEEdit do Html tabuľky, podobne ako property AddTableRow, TableBegin a TableEnd.

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

Label + TextBox + Label alebo Button

Ak property AddTdToLabel nastavíme na hodnotu True (property AddTableRow musí byť nastavená na False), potom sa nám medzi tieto prvky komponenty PxEEdit vložia elementy tabuľky nasledovne:

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

Ak property AddOnlyTdToLabel nastavíme na hodnotu True (property AddTableRow, AddTdToLabel musia byť nastavená na False), potom sa nám medzi tieto prvky komponenty PxEEdit vloží element tabuľky td nasledovne:

Label + <td>TextBox

Property AddTdToLabel, AddOnlyTdToLabel sú implementované aj ďalších v komponentách, ako sú PxComboBox, PxMemo, PxJSDatePicker, PxCheckBox, PxCheckBoxList a pod.

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

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

Zmena popisku komponenty PxEEdit (Label):

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

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

Zmena popisku At Post komponenty PxEEdit (Label):

Komponenta PxEEdit obsahuje ešte ďalší popisok, ktorý sa nachádza za komponentov TextBox. Ku tomuto popisku sa dá prístupit' 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:

Editovať adresu

Ok Storno

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

Komponenta PxComboBox pracuje obdobne, alebo ma podobné property, ako komponenta PxEEdit. Preto nemá zmysel sa nimi podrobne zaoberať, môžete si prečítať kapitolu predchádzajúcu o komponente PxEEdit. Veci, ktoré boli podrobne popísané pri komponente PxEEdit, budú tuá 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 informácií 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.ConnectionString = "User Id=database01;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 vhodné 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.

Editovať adresu

ID	2
Meno Osoby	Ján Kolenička
Adresa	Nitra
Právna Forma	Fyzická osoba
Dátum založenia	Zvoľte hodnotu Fyzická osoba Právnická osoba
Ok Storno	

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 centrálné, 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 prísť 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:

Editovať adresu

The screenshot shows a form titled "Editovať adresu". It contains the following fields and values:

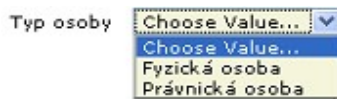
- ID: 2
- Meno Osoby: Ján Kolenička
- Adresa: Nitra
- Právna forma 2: Fyzická osoba (The label "Právna forma 2" is circled in red, and the dropdown value "(Text)" is also circled in red.)
- Dátum založenia: 20.02.2010

At the bottom of the form are two buttons: "Ok" and "Storno".

Ak pri komponente *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 tuá 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 *PxCheckBox*:

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 možné 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 vyzerat' nasledovne:

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

Tuto definíciu potom môžeme zapísať do súboru Adresar.aspx.cs, a bude vyzerat' 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.ConnectionString = "User Id=database01;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 zapísaní, 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 Mőciková	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 formulári AddAdresar.aspx to bude vyzerat' nasledovne:

Editovať adresu

ID

Meno Osoby

Adresa

Právna Forma

Dátum založenia 

☐ Adresa neplatná

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_address"].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:

Editovať adresu

ID	13
Meno Osoby	Helena Mőciková
Adresa	Horné Lužné
Právna Forma	Fyzická osoba
Dátum založenia	10.03.2010
	☐ Adresa neplatná 2
Ok Storno	

1.7. PxFlyComboBox – séria comboboxov, zapojených za sebou, vhodný pre prácu so 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 ľubovoľný počet 2 až n. Viac obrázkov:

Kraj	Okres	Obec
Banskobystrický	Banská Bystrica	Brusno

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 zvolený 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 jedinečné, 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 zobrazit' 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 zobrazit' 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 zobrazit' 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 zobrazit' v komponente PxSuperGrid. Jednotlivé názvy stĺpcov sú oddelené bodkočiarkou.

aPxWebQueryName – Názov komponenty PxWebQuery, v ktorej je natiiahnutý 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 zadaného 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 vyzerat' nasledovne:

Obec, Okres
Korytné, Levoca
Banská Bystrica, Banská Bystrica
Cucma, Rožnava

Pri zadaní reťazca "Obec;Okres" do položky FieldToView, nastavenom móde zobrazenia **mvValueBR**, bude

zobrazenie v komponente PxSuperGrid vyzerat' nasledovne:

Obec, Okres
Korytné Levoca
Banská Bystrica Banská Bystrica
Cucma Rožnava

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

Obec, Okres
Obec: Korytné Okres: Levoca
Obec: Banská Bystrica Okres: Banská Bystrica
Obec: Cucma Okres: Rožnava

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

Obec, Okres
Obec: Korytné Okres: Levoca Kraj: Prešovský
Obec: Banská Bystrica Okres: Banská Bystrica Kraj: Banskobystrický
Obec: Cucma Okres: Rožnava Kraj: Košický

Celkové zobrazenie komponenty PxSuperGrid bude vyzerat' nasledovne:

...	ID	Meno Osoby	Ulica	Právna Forma	Obec, Okres
   	14	JUraj Handli	Severna 7	Fyzická osoba	Obec: Korytné Okres: Levoca Kraj: Prešovský
   	15	Helena Môciková	THK 3	Fyzická osoba	Obec: Banská Bystrica Okres: Banská Bystrica Kraj: Banskobystrický
   	16	Jana Mikletič	Šalgotarianska 22	Fyzická osoba	Obec: Cucma Okres: Rožnava Kraj: Košický
 PRIDAŤ NOVÝ RIADOK...					
Page 1					

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 novú komponentu PxWebQuery, a tu nazveme wquObec. V súbore Adreasar.aspx doplníme túto novú komponentu, a definícia bude vyzeráť nasledovne:

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

Nezabudnime komponentu wquObec, zdefinovať 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 definovaní 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	Veľká Hradná	24	Trencín	3	Trenciansky
3	Veľké Bierovce	24	Trencín	3	Trenciansky
4	Višňové	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.ConnectionString = "User Id=database01;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.ConnectionString = "User Id=database01;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 zdefinujeme 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 vyzeráť 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 PxEEdit 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 vyzerat nasledovne:

Editovať adresu

ID

Meno Osoby

Právna Forma

Dátum založenia

☐ Adresa neplatná

Ulica

Kraj Okres Obec

PxFlyComboBox

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:

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

Komponenta *PxFlyComboBox*, by vyzerala nasledovne:

Kraj Okres Obec

Banskobystrický

Bratislavský

Košický

Nitriansky

Prešovský

Trenčiansky

Trnavský

Žilinský

Komponenta *PxFlyComboBox* obsahuje funkciu cez ktorú môžeme zistiť nastavenú hodnotu jednotlivých comboboxov. Táto metóda sa nazýva **GetValue**.

Popis Parametrov funkcie GetValue:

C# syntax:

```
public string GetValue(string aFieldName);
```

Popis parametrov:

FieldName – Meno stĺpca ktorý prislúcha ku danému comboboxu v komponente PxFlyComboBox. Obyčajne sú to mená stĺpcov, ktoré boli zadane v parametre AddParamFlyComboBox, a to tie ktoré boli zadane do výrazu **aKeyFields**, **aKeyDetailFields** a **aValueFields**.

Príklad použitia metódy GetValue:

```
string sGetVal = fcmbMiesto.GetValue("IdOkres");
```

V tomto príklade nám vráti funkcia GetValue hodnotu nastaveného okresu, ktorý je nastavený v comboboxe, v ktorom sa vyberá zo zoznamu daný okres.

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:



Spojenie	
Telefón	0905 123 456
Poznámka	mobil

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

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 jedinečné, v tabuľke nesmie existovať stĺpec s týmto menom).

aEditAllFieldNames – 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.

aExpresionToView – 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 zobrazení 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 zadaného 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 vyzeráť nasledovne:

Spojenie
0905 123 456 - mobil 02/456455 - pevna linka
0905 123 456 - mobil

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

Spojenie
0905 123 456 - mobil
02/456455 - pevna linka
0905 123 456 - mobil

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

Spojenie	
1.	0905 123 456 - mobil
2.	02/456455 - pevná linka
1.	0905 123 456 - mobil

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 zobrazení v komponente PxSuperGrid-u. Ak v property VisibleFields, nie je priradená 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 zdefinovať 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 vyzeráť nasledovne:

...	Meno Osoby	Ulica	Právna Forma	Spojenie	Obec, Okres
 	Peter Tkáč	Kolkareň 13/18	Fyzická osoba	1. 0905 123 456 - mobil 2. 02/456455 - pevná 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.ConnectionString = "User Id=database01;Password=aa;Data Source=xe;";
    wquAdresar.SQLSelect = @"select idadresar, name, address, idpravnaforma,
        create_date, invalid_address, 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_address", "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_address"].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="fembMiesto" 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 vyzerat' 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 tag
.

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 vyzerat' nasledovne:

Editovať adresu

ID	4	
Meno Osoby	Helena Môciková	
Právna Forma	Fyzická osoba	
Dátum založenia	03.03.2010	
	☐ Adresa neplatná	
Ulica	Šalgotarianska 22	
Kraj	Okres	Obec
Trnavský	Piešťany	Šípkové

+

Spojenie

Telefón

Poznámka

+

Fax

Fax

Poznámka

Ok

Storno

V komponente PxGreatRepeater sú integrované komponenty PxComboBox, PxFlyComboBox, PxJSDatePicker, a pri uvalení 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:

+

Kompetencie

Špecializácia

Číslo osvedčenia

Dátum vydania osvedčenia

+

Názov Obce, Okres

Ulica, číslo

Kraj

Okres

Obec

Komponenta PxGreatRepeater obsahuje funkciu cez ktorú môžeme zistiť nastavenú hodnotu

jednotlivých komponent, ktoré obsahuje komponenta PxGreatRepeater. Táto metóda sa nazýva **GetValue**.

Popis Parametrov funkcie GetValue:

C# syntax:

```
public string GetValue(string aFieldName, Int32 aNumberRepeaterPanel);
```

Popis parametrov:

aFieldName – Meno stĺpca ktorý prislúcha ku danej komponente, ktorá je obsiahnutá v komponente PxGreatRepeater. Obyčajne sú to mená stĺpcov, ktoré boli zadané v parametre **AddParamGreatWebQuery**, a to tie ktoré boli zadané do výrazu **aEditAllFieldNames**. Pri komponente PxFlyComboBox, ktorá sa nachádza v komponente PxGreatRepeater, dochádza ku preťaženiu premennej aFieldName, a to spôsobom, že sa zadá najprv FieldName, ktorý definuje komponentu PxFlyComboBox a potom sa zadá meno fieldu daného comboboxu, oddeleného bodkočiarkou. Viac príklad: "Miesto1;IdOkres1"

aNumberRepeaterPanel – číslo panela z ktorého sa má načítať hodnota. Číslo prvého panela je 0. Maximálny počet panelov je obmedzený premennou **aMaxRepeatCount** v parametre **AddParamGreatWebQuery**.

Príklad použitia metódy GetValue:

```
string sGetVal = gtrTelefon.GetValue("Tell", 0);
```

V tomto príklade nám vráti funkcia GetValue hodnotu nastavenú v položke "Tell", v prvom paneli.

Príklad použitia metódy **GetValue** a načítanie hodnoty z komponenty PxFlyComboBox, ktorá sa nachádza v komponente PxGreatRepeater:

```
string sGetVal = gtrMiesto.GetValue("Miesto1;IDOBEC1", 0);
```

V tomto príklade nám vráti funkcia GetValue hodnotu nastaveného okresu komponenty PxFlyComboBox, ktorá je vložená v komponente PxGreatRepeater.

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 stlačení 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` zadať nasledovne:

```
<Px: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 = "FRNLIEDPCH";
```

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`
H – (Refresh) znovu načítanie, občerstvenie dát z SQL databázy

Tu v ďalšom sú popísané obrázky tlačítok, ktoré prislúchajú jednotlivým operáciám, 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“



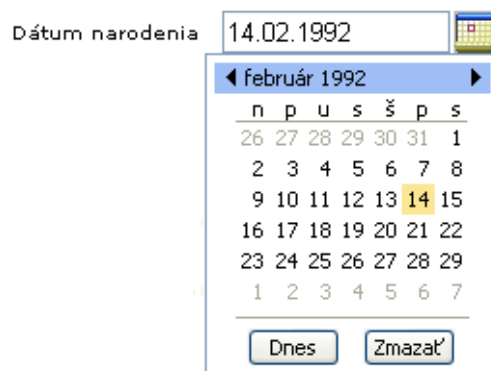
- Operácia Cancel, do property `PxVisibleButtons` zapíšeme písmeno „C“



- Operácia Refresh, do property `PxVisibleButtons` zapíšeme písmeno „H“

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 = "CREATE_DATE";
```

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

Pokiaľ by sme chceli zmeniť formát dátumu, môžeme to urobiť cez property DateFormat, viac príklad dole:

```
edtCREATE_DATE.DateFormat = "d.M.yyyy HH:mm:ss";
```

Zadanie tohto formátu, sa nám na komponente PxJSDatePicker prejaví nasledovne, viac na obrázku dole:

Dátum založenia 7.7.2010 12:00:00

1.11. PxDatePicker – komponenta pre zadanie dátumu

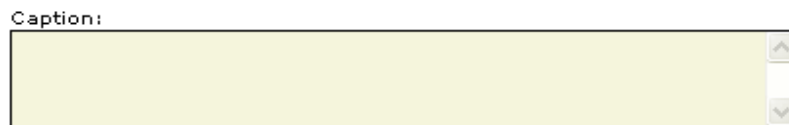
Komponenta PxDatePicker bola z Px Frameworku vo verzii v.1.08.17 vyradená. Preto doporučujeme používať komponentu PxJSDatePicker, ktorá túto komponentu nahradzuje.

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

Komponenta PxMemo je obdobou komponenty PxEEdit, a preto nemá veľký zmysel sa zaoberať popisom komponenty PxMemo, pokiaľ chcete vedieť viac o komponente PxMemo, vráťte sa ku kapitole, o komponente PxEEdit. Komponenta PxMemo je obdobou komponenty PxEEdit, a v časti OnInit, sú pridané navyše tieto property, ktoré menia PxEEdit na komponentu 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 zdefinovať 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 PxEEdit.

1.13. PxLabel – komponenta pre zobrazenie dát

Komponenta PxLabel slúži na zobrazenie dát prostredníctvom komponenty PxWebQuery. Dole na

obrázku je jej zobrazenie:

Telefón 02/454545

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 centrálnne, 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 vyzerat' nasledovne:

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

```
PxInfoFormName="InfoAdresar.aspx"  
PxVisibleButtons="SIDE0">  
<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 vyzerat’ 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="lblICREATE_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="fembMiesto" 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="Zavriet" 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");
}

```

1.14. PxCheckBoxList – komponenta pre zobrazenie a výber hodnôt zo zoznamu

Komponenta PxCheckBoxList má mnoho podobných 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 tuá prejdené len telegraficky, zbežne.

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

Súbor AddAdresar.aspx:

```
<Prx:PxCheckBoxList ID="chlOrgan" runat="server" AddTableRow="True" />
```

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

Súbor AddAdresar.aspx.cs:

```
protected void Page_Load(object sender, EventArgs e)
{
    chlOrgan.PxDataSource = wquAdresar;
    chlOrgan.FieldName = "Z_REZORTNYORGAN";
}
```

Komponenta *PxCheckBoxList* podporuje automatické načítanie hodnoty z komponenty *PxWebQuery*, a automatické uloženie zvolenej hodnoty do databázy, pri vykonaní príkazu *Post()* na komponente *PxWebQuery*. Tým že komponenta *PxCheckBoxList* umožňuje, vyberať jednotlivé položky zo zoznamu, je možné zadať na komponente *PxWebQuery*, ako tieto stavy budú zobrazované a zapísané do databázy. Na definíciu sa používa parameter s názvom metódy *AddParamCheckList*. Metóda *AddParamCheckList* je metódou komponenty *PxWebQuery*.

Popis Parametrov metódy *AddParamCheckList*:

C# syntax:

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

public void AddParamCheckList(string aFieldName, string aFieldNameKey, string
aFieldNameValue, string aPxWebQueryName)
```

Popis parametrov:

aFieldName – Meno stĺpca v tabuľke na ktorý sa aplikuje metóda *AddParamCheckList*. V databáze je najlepšie priradiť tomuto stĺpcu veľkosť *varchar(500)*

aFieldNameKey – Meno stĺpca číselníka, ktorý reprezentuje kľúč 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)*, pri výbere viacerých položiek, sa tieto položky pri zobrazení v komponente *Memo(TextBox)*, budú zobrazovať oddelené čiarkou

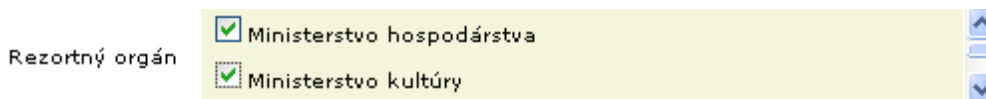
aFieldToDB - tu sa zadáva meno stĺpca, ktorého hodnota bude zapísaná do databázy, pri výbere viacerých položiek, sa tieto položky zapisujú jedna za druhou a budú oddelené čiarkou

aPxWebQueryName – Názov komponenty *PxWebQuery* v ktorej je cez *SQL* príkaz načítaný číselník, jednotlivé stĺpce je vhodné označiť ako *Key* a *Value*, kvôli lepšej prehľadnosti

Príklad zadania v súbore AddAdresar.aspx.cs v metóde *Page_Load*:

```
wquAkcia.AddParamCheckList("Z_REZORTNYORGAN", "IDREZORTNYORGAN",
"NAZOVREZORTNYORGAN", "wquRezortOrg");
```

Po uvalení parametra *AddParamCheckList* na komponentu *PxWebQuery*, a napojení komponenty *PxCheckBoxList*, 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["REZORTNYORGAN"].Caption = "Rezortný orgán";
```

Zmena popisku komponenty `PxCheckBoxList` (Label):

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

```
chlOrgan.Caption = "Rezortný orgán 2";
```

Komponenta `PxCheckBoxList` má property s názvom *OnlyCheckBoxList*, ktorá umožňuje zobrazovať komponentu `PxCheckBoxList` dvoma spôsobmi, pokiaľ nie je v stave editácie. Pokiaľ je do property *OnlyCheckBoxList* zadaná hodnota *True*, potom je komponenta zobrazovaná ako `CheckBoxList`, podobne ako v stave editácie. Pokiaľ je do property *OnlyCheckBoxList* zadaná hodnota *False*, potom je komponenta zobrazovaná ako `Memo(TextBox)`, a jednotlivé zvolené hodnoty sú zapísané za sebou a sú oddelené čiarkou. Viac obrázkov dole.



```
chlOrgan.OnlyCheckBoxList = false;
```

Cez property *SeparatorSumText* komponenty `PxCheckBoxList`, môžeme ovládať zobrazenie oddeľovača hodnôt pri zobrazení v komponente `Memo(TextBox)`, a pri zapísaní do databázy. Defaultna hodnota je čiarka `“,”`, poprípade môžete zadať bodkočiarku `“;”`.

```
chlOrgan.SeparatorSumText = ",";
```

Komponenta `PxCheckBoxList` umožňuje ďalej filtrovanie zobrazeného obsahu číselníka cez property *Filter*. V property *Filter* sa používajú názvy stĺpcov(`fieldov`), číselníka komponenty `PxWebQuery`.

```
chlOrgan.Filter = "Aktualny_Stav=1";
```

Po zadání tohto príkazu sa nám v komponente `PxCheckBoxList`, zobrazia len tie položky, ktoré majú v stĺpci číselníka komponenty `PxWebQuery` s názvom `„Aktualny_Stav“` zadanú hodnotu `„1“`.

1.15. PxRadioButtonList – komponenta pre zobrazenie a výber hodnoty zo zoznamu

Komponenta *PxRadioButtonList* má mnoho podobných 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 tu napríklad len telegraficky, zbežne.

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

Súbor *AddAdresar.aspx*:

```
<Prx:PxRadioButtonList ID="rblPravoplatRozhodnutie" runat="server" AddTableRow="false"
CaptionOnTop="True" Width="150" />
```

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

Súbor *AddAdresar.aspx.cs*:

```
protected void Page_Load(object sender, EventArgs e)
{
    rblPravoplatRozhodnutie.PxDataSource = wquAdresar;
    rblPravoplatRozhodnutie.FieldName = "PravoplatRozhodnutie";
}
```

Komponenta *PxRadioButtonList* podporuje automatické načítanie dát z komponenty *PxWebQuery*, a automatické uloženie zvolenej hodnoty do databázy, pri vykonaní príkazu *Post()* na komponente *PxWebQuery*. Tým že komponenta *PxRadioButtonList* umožňuje, vybrať len jednu položku zo zoznamu, je možné zadať na komponente *PxWebQuery*, ako tieto stavy budú zobrazované a zapísané do databázy. Na definíciu sa používa parameter s názvom metódy *AddParamRadioList*. Metóda *AddParamRadioList* je metódou komponenty *PxWebQuery*.

Popis Parametrov metódy *AddParamRadioList*:

C# syntax:

```
public void AddParamRadioList(string aFieldName, string aDataKey, string
aDataValue, string aFieldToView, string aFieldToDB)
```

Popis parametrov:

aFieldName – Meno stĺpca v tabuľke na ktorý sa aplikuje parameter *AddParamRadioList*

aDataKey – Meno stĺpca do ktorého sa zadá reťazec kľúčov, oddelených bodkočiarkou, v tomto prípade to je „1;2“

aDataValue – Meno stĺpca do ktorého sa zadá reťazec hodnôt (value), oddelených bodkočiarkou, v tom

poradí ako boli zadané hodnoty kľúčou, v tomto prípade to je „Posudzovať;Neposudzovať“. Tento text bude aj zobrazený pri jednotlivých butónoch v komponente PxRadioButtonList

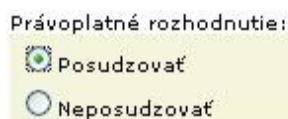
aFieldToView – tu sa zadáva meno stĺpca, ktorého hodnota sa bude zobrazovať v Gride(*PxSuperGrid*), v tomto prípade sú to texty „Value“ a „Key“.

aFieldToDB - tu sa zadáva meno stĺpca, ktorého hodnota bude zapísaná do databázy, v tomto prípade sú to texty „Value“ a „Key“.

Príklad zadania v súbore AddAdresar.aspx.cs v metóde Page_Load:

```
wquAkcia.AddParamRadioList("PravoplatRozhodnutie", "1;2",  
"Posudzovať;Neposudzovať", "Value", "Key");
```

Po uvalení parametra AddParamRadioList na komponentu PxWebQuery, a napojení komponenty PxRadioButtonList, si táto komponenta pri načítaní automaticky natiahne dáta z parametra, 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["PravoplatRozhodnutie"].Caption =  
"Právoplatné rozhodnutie:";
```

Zmena popisku komponenty PxRadioButtonList (Label):

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

```
rblPravoplatRozhodnutie.Caption = " Právoplatné  
rozhodnutie 2";
```

Pokiaľ by sme chceli umiestniť popisok komponenty PxRadioButtonList nad telo komponenty(tak ako je to na hornom obrázku), použijeme k tomu property *CaptionOnTop*, viac príklad dole:

```
rblPravoplatRozhodnutie.CaptionOnTop = true;
```

Komponenta PxRadioButtonList bola vybavená navyše nulovacím tlačítkom, ktoré sa aktivuje nastavením property *ThreeState* na hodnotu „true“. Nulovacie tlačítko slúži na vynulovanie obsahu komponenty PxRadioButtonList.

```
rblPravoplatRozhodnutie.ThreeState = true;
```

To ako vyzerá komponenta PxRadioButtonList s nulovacím tlačítkom, vidíte na obrázku dole.

Právoplatné rozhodnutie:

☐ Posudzovať
☒ Neposudzovať



1.16. PxChart – komponenta pre zobrazenie a prácu s grafmi

Komponenta PxChart je komponentou Px Frameworku, prostredníctvom ktorej môžete vytvárať atraktívne 2D a 3D grafy pre vaše webové aplikácie. Komponenta PxChart môže priamo načítať dáta z komponenty PxWebQuery, alebo priamo z XML súboru. PxChart obsahuje širokú škálu grafov, ako tradičný Bar graf, Line, Area, koláč a pod.

Na správnu funkciu komponenty PxChart je potrebné, aby ste mali nainštalovaný Flash Player vo vašom webovom prehliadači.

Pokiaľ chceme pracovať s komponentou PxChart, musíme si ju v súbore *.aspx zadať nasledovne:

```
<Prx:PxChart ID="Chart" runat="server"></Prx:PxChart>
```

Ak chceme, aby komponenta PxChart bola viditeľná v aspx súboroch, musíme si v hlavičke súboru zadať "Prx" prefix s prepojením na assembly komponenty PxChart. Toto vykonáme nasledovne:

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

Keď máme toto vykonané, môžeme v súbore *.aspx.cs s komponentou PxChart pracovať nasledovne. Najprv si do komponenty PxWebQuery načítame dáta z databázy, a potom ju prepojíme z komponentou PxChart. Najprv ku komponente PxChart cez property *PxDataSource* pripojíme komponentu PxWebQuery. Potom cez property *FieldName* a *FieldValue* komponenty PxChart, zadefinujeme prepojenie stĺpcov komponenty PxWebQuery na jednotlivé osi, X a Y komponenty PxChart. Viac príklad dole:

```
Chart.PxDataSource = wquStatistics;  
Chart.FieldName = "name";  
Chart.FieldValue = "pocet";
```

Popis grafu môžeme zadať cez property *Caption* a *SubCaption*, viac príklad dole:

```
Chart.Caption = "Štatistika žánrov";  
Chart.SubCaption = "Sumár žánrov za všetky CD tituli";
```

Podobne si môžeme zadať typ grafu cez property *TypeChart*. Jednotlivé typy grafů sú zadefinované vo výčtovom type *PxChart.enumTypeCharts*. Viac príklad dole:

```
Chart.TypeChart = PxChart.PxChart.enumTypeCharts.tcColumn3D;
```

Celý zdrojový kód príkladu ktorý sme si tu preberali, aj s načítaným dát do komponenty *PxWebQuery* a prepojením na komponentu *PxChart* a výberom typu grafu, nájdete dole, v priloženom výpise zdrojového kódu.

```
protected void Page_Load(object sender, EventArgs e)
{
    if (wquStatistics.Active == false)
    {
        wquStatistics.ConnectionString = AppConst.ConnectionString;
        wquStatistics.SQLSelect = @"select CdTitul.IdZaner as idzaner,COUNT(*) as pocet,
                                   Zaner.Name as name from CdTitul, Zaner
                                   where CdTitul.IDZaner=Zaner.IdZaner
                                   group by CdTitul.IDZaner";

        wquStatistics.Open();
    }
    Chart.PxDataSource = wquStatistics;
    Chart.FieldName = "name";
    Chart.FieldValue = "pocet";

    Chart.Caption = "Štatistika žánrov";
    Chart.SubCaption = "Sumár žánrov za všetky CD tituli";
    //Chart.Width = 800;
    //Chart.Height = 800;

    Chart.TypeChart = PxChart.PxChart.enumTypeCharts.tcColumn3D;
    Chart.AddHeader("canvasBgColor", "F6DFD9");
    Chart.AddHeader("divlinecolor", "F47E00");
}
```

Cez metódu *AddHeader*, môžeme do komponenty *PxChart* vkladať rôzne elementy aj s definíciou hodnoty, ktoré môžu rôzne ovplyvniť formu a tvar výstupného grafu.

Popis parametrov metódy *AddHeader*:

C# syntax:

```
public void AddHeader(string Tag, string Value)
```

Popis parametrov:

Tag – do tejto premennej sa zadáva názov (name) tágu

Value – do tejto premennej sa zadáva hodnota(value) tágu

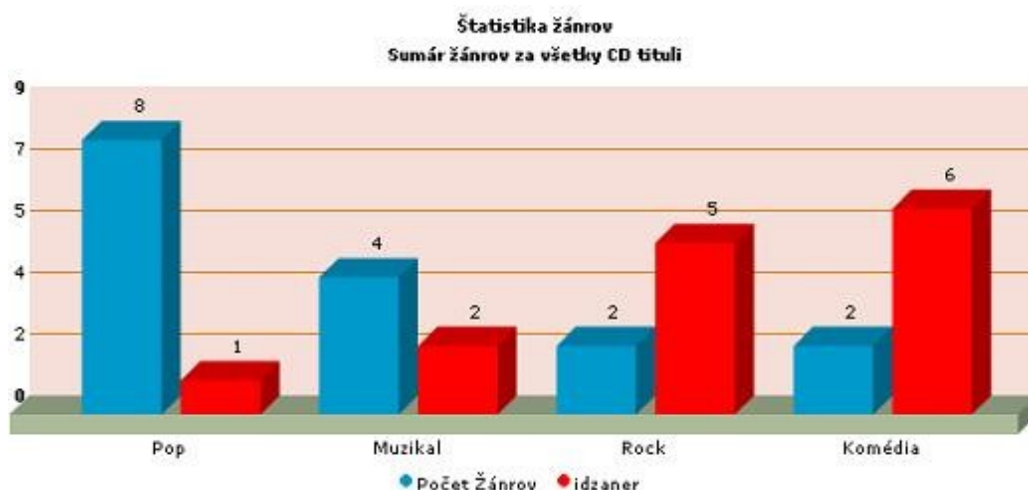
Obrázok výsledného grafu vyzerá nasledovne, viac obrázkov dolu:



Pokiaľ by sme chceli pracovať so sériovým grafom, t.j. viac inštancií grafov vedľa seba, môžeme to urobiť cez property *FieldValue* komponenty PxChart. Urobíme takzvané preťaženie property *FieldValue*, a to tým že do tejto property zadáme za sebou viac názvov stĺpcov z komponenty PxWebQuery, ktoré sú oddelené bodkočiarkou. Ešte samozrejme do property *TypeChart*, musíme vybrať nejaký sériový graf, napr. tcMSColumn3D. Každý sériový graf sa začína prefixom tcMS... Viac príklad dole:

```
Chart.PxDataSource = wquStatistics;
Chart.FieldName = "name";
Chart.FieldValue = "pocet;idzane";
Chart.TypeChart = PxChart.PxChart.enumTypeCharts.tcMSColumn3D;
```

Obrázok výsledného grafu vyzerá nasledovne, viac obrázkov dolu:



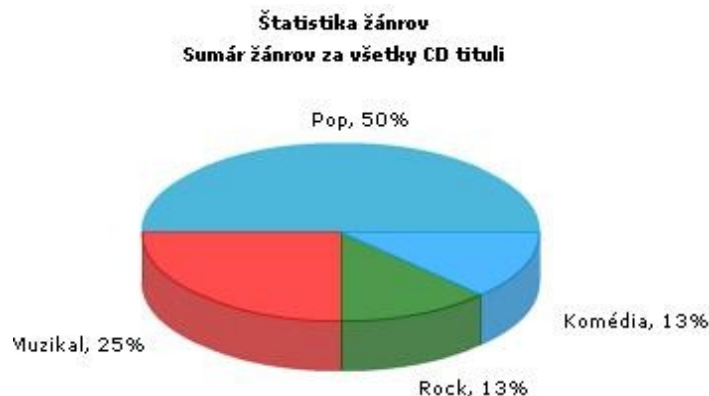
Pokiaľ by sme chceli graf, takzvaný koláčik, môžeme to urobiť zadaným typom grafu tcPie3D do property *TypeChart*. Viac príklad dole:

```

Chart.PxDataSource = wquStatistics;
Chart.FieldName = "name";
Chart.FieldValue = "pocet";
Chart.TypeChart = PxChart.PxChart.enumTypeCharts.tcPie3D;

```

Obrázok výsledného grafu vyzerá nasledovne, viac obrázkov dolu:



Pokiaľ by ste chceli načítať dáta do komponenty PxChart zo súboru xml a nie z komponenty PxWebQuery, môžete tak urobiť cez property *ChartXML*. V prípade že načítate dáta do komponenty PxChart cez property *ChartXML*, všetky property komponenty PxChart budú zablokované, okrem property *TypeChart*, do ktorej sa zadáva typ grafu. Vzor XML súboru by mohol vyzerat' nasledovne, viac príklad dole:

```

<graph caption='Štatistika žánrov' subCaption='Sumár žánrov za všetky CD tituli'
showPercentageInLabel='1' pieSliceDepth='25' decimalPrecision='0' showNames='1'
pieRadius='100' >
  <set name='Pop' value='8' color='0099CC' />
  <set name='Muzikal' value='4' color='FF0000' />
  <set name='Rock' value='2' color='006F00' />
  <set name='Komédia' value='2' color='0099FF' />
</graph>

```

V časti *<graph>* zadávame elementy ktoré bližšie popisujú výsledný graf. Sú to napr. tágy *Caption*, *Subcaption*, *showNames* a pod. V časti *<set>* zadávame meno a hodnotu výstupného grafu. Komponenta PxChart je zapuzdreným free komponenty FusionCharts v.2.0. Bližší popis jednotlivých elementov ktoré majú vplyv na zobrazenie grafu a popis vytvárania výsledného XML súboru môžete nájsť v dokumentácii ku komponentám FusionCharts v.2.0, alebo na internetovej adrese: <http://www.proxima-soft.sk/data/FusionChartsFree/Contents/menu.html>

1.17. PxFilterView – vizuálna komponenta pre filtrovanie obsahu dát tabuľky v komponente PxWebQuery

S pomocou komponenty PxFilterView môžeme filtrovať obsah dát tabuľky v komponente PxWebQuery.

V súbore *.aspx sa komponenta PxFilterView definuje nasledovne:

```
<Prx:PxFilterView ID="filterSposOsoba" runat="server" />
```

Do komponenty PxFilterView zadávame parametre, výberové kritéria cez metódu AddParamFilter. Podľa týchto parametrov sa nám dynamicky zostavuje komponenta PxFilterView. Viac obrázok dolu:

Typ Osoby :	Fyzická osoba ▼
Kraj:	Košický ▼
Okres:	Košice II ▼
Obec:	Košice - Pereš ▼
Špecializácia:	Zvoľte hodnotu ▼
Hľadať reťazec:	ján

Do takto vykreslenej komponenty PxFilterView, zadávame výberové kritéria, z ktorých sa nám zostaví výsledný SQLScript, ktorý je potom priradený do komponenty PxWebQuery.

Na to aby sa nám vykreslil nasledujúci PxFilterView, v takejto podobe ako je na obrázku musíme zadať nasledujúce parametre:

```
filterSposOsoba.AddParamFilter("IDTypOsoba", "<b>Typ Osoby:</b>");  
filterSposOsoba.AddParamFilter("Miesto", "<b>Kraj:</b>", "Kraj1");  
filterSposOsoba.AddParamFilter("Miesto", "<b>Okres:</b>", "Okres1");  
filterSposOsoba.AddParamFilter("Miesto", "<b>Obec:</b>", "Obec1");  
filterSposOsoba.AddParamFilter("Specializacia", "<b>Špecializácia:</b>", "SPECIALIZATION1");  
filterSposOsoba.AddParamFilter("FIRST_NAME;LAST_NAME", "<b>Hľadať reťazec:</b>");  
  
filterSposOsoba.OrigSQLSelect = "select * from SposOsoba";
```

Na konci metódy „Page_Load“, nesmieme ešte zabudnúť prepojiť komponentu PxFilterView s komponentou PxWebQuery, do ktorej sa nám bude zadávať nový SQL príkaz vygenerovaný komponentou PxFilterView:

```
filterSposOsoba.PxDataSource = wquSposOsoba;
```

Definícia metódy AddParamFilter:

```
public void AddParamFilter(string aFieldName, string aCaption)
```

```
public void AddParamFilter(string aFieldName, string aCaption, string aParentFieldName)
```

```
public void AddParamFilter(string aFieldName, string aCaption, string aParentFieldName, string aChooseValueText)
```

Popis parametrov:

aFieldName – Meno stĺpca tabuľky komponenty PxWebQuery, podľa ktorého chceme filtrovať obsah dát tabuľky komponenty PxWebQuery

aCaption – Popisok, titulok riadkov komponente PxFilterView

aParentFieldName – tento stĺpec sa vyplňa len vtedy, keď na stĺpec definovaný v parametre aFieldName bol uvalený parameter AddParamFlyComboBox alebo AddParamGreatWebQuery. Treba si uvedomiť že pokiaľ sú

na nejaký `aFieldName` uvalené nasledovné parametre `AddParamFlyComboBox` alebo `AddParamGreatWebQuery`, tak tento field, stĺpec je zlučovaci, t.j. môže sa pod ním ukrývať mnoho fieldov. Takýto field môžeme nazvať ako child, a stĺpce ktoré zlučuje, ako Parent. Do parametra `aParentFieldName` zadáme konkrétne meno parent fieldu, podľa ktorého chceme filtrovať obsah dát tabuľky `PxWebQuery` **aChooseValueText** – do tohto parametra zadávame text, ktorý sa nám bude zobrazovať v komponente `PxFilterView`, u konkrétneho riadku zobrazenia, pokiaľ vyber bude vykonávaný v zozname(`ComboBox`). Tento text nám nahradí text “Zvoľte hodnotu”. Viac obrázkov hore.

Komponenta `PxFilterView` pracuje tak, že pri výbere zo zoznamu alebo zadaný nejakého textu a potvrdený klávesou `Enter`, sama generuje SQL príkaz. Vzhľadom na to že je pripojená ku konkrétnej komponente `PxWebQuery`, pri každej zmene je tento nový SQL príkaz priradený do komponenty `PxWebQuery` a je vykonaný `ReOpen` komponenty `PxWebQuery`. Takto pri zmene výberových kritérií komponenty `PxFilterView`, sa aj zmení obsah dát tabuľky `PxWebQuery`.

Ak by s me mali stĺpce v komponente `PxWebQuery`, na ktoré nie uvalený žiaden parameter (`AddParamKey`, `AddParamWebQuery` a pod.) a sú tieto stĺpce typu `string`, môžeme pri zadávaní parametra tieto stĺpce zlúčiť a oddeliť ich bodkočiarkou. Pri vyhľadávaní sa vložený výraz vyhľadáva v obidvoch týchto zadaných stĺpcoch. Viac príklad dole:

```
filterSposOsoba.AddParamFilter("FIRST_NAME;LAST_NAME", "<b>Hľadať reťazec:</b>");
```

Potom keď do filtra `PxFilterView`, do riadku s popiskom „Hľadať reťazec“, zadáme nejaké slovo, toto slovo bude vyhľadávané v obidvoch stĺpcoch, a tak bude vyfiltrovaná konečná podoba tabuľky.

Je zaujímavé podľa akých kritérií zostavuje komponenta `PxFilterView` svoju editačnú časť? Pre stĺpce, na ktoré nebol uvalený žiadny parameter, sa v editačnej časti zjaví komponenta `TextBox`, do ktorej sa zadáva text, ako výberové kritérium. V prípade že na daný stĺpec bol uvalený parameter `AddParamKey`, `AddParamWebQuery` a pod. je komponenta `TextBox` nahradená komponentou `DropDownList`, aj s naplneným listom, obsahom tejto komponenty. Komponenta `PxFilterView` dokáže čítať obsah tabuliek a ich vzájomného prepojenie(jednotlivé parametre), a na základe toho zkompiluje výsledný panel komponenty `PxFilterView`.

Len na nás zaleží či dokážeme správne vyplniť všetky premenné ktoré sa zadávajú cez metódu `AddParamFilter`.

Komponenta `PxFilterView`, okrem klasického zadávania parametrov cez `AddParamFilter()`, podporuje aj zadávanie ďalších parametrov, ktoré sa prikompilujú ku výslednému `Sql` príkazu, čím je možné obslužiť i zložitejšie požiadavky na filtrovanie prostredníctvom komponenty `PxFilterView`. Zadávanie ďalších parametrov sa robí cez metódu `RebuildSqlSelect()`.

Popis Parametrov metódy `RebuildSqlSelect`:

C# syntax:

```
public void RebuildSqlSelect(string AddSelect, string AddFrom, string AddWhere)
```

Popis parametrov:

AddSelect – Reťazec, ktorý sa pridá do časti medzi **select** a **from** SQL príkazu

AddFrom – Reťazec, ktorý sa pridá do časti medzi **from** a **where** SQL príkazu

AddWhere – Reťazec, ktorý sa pridá do časti za **where** SQL príkazu

Metóda *RebuildSqlSelect()* má pamäťový efekt, teda komponenta *PxFilterView* si zapamätá posledne zadaný stav, a pri každej zmene komponenty *PxFilterView* sa prikompilujú tieto zadané a zapamätané parametre k výslednému Sql príkazu, ktorý sa potom zapíše do komponenty *PxWebQuery* a vyvolá sa následne procedúra *ReOpen()* tejto komponenty. Ak chceme vymazať tieto parametre z komponenty *PxFilterView*, musíme zavolať metódu *RebuildSqlSelect()* ešte raz s prázdnyimi parametrami.

Tu v ďalšom si uvedieme príklad použitia metódy *RebuildSqlSelect()*.

Do súboru *.aspx ku definícii komponenty *PxFilterView* sme si zadali, pod filter ďalší prvok, ktorý bude súčasťou filtra komponenty *PxFilterView*.

```
<Prx:PxFilterView ID="filterSposOsoba" runat="server"
    TableEnd="false" />
<tr><td></td><td>
    <asp:CheckBox ID="chkFyzickaOsoba" Text="Zobraziť len fyzickú
        osobu" runat="server" Checked="false" AutoPostBack="true"
        oncheckedchanged="chkFyzickaOsoba_CheckedChanged" />
    </td></tr>
</table>
```

Tento ďalší prvok je komponenta *CheckBox*, pri ktorej zaškrtnutí by sa nám vyfiltrovala, len fyzická osoba zo zoznamu. Táto komponenta má nastavený *AutoPostBack* na hodnotu *true*, čo znamená, že pri zmene jej stavu, sa zavolá metóda zapísaná v property *oncheckedchanged*, ktorá sa volá „chkFyzickaOsoba_CheckedChanged“. Telo tejto metódy vyzerá nasledovne:

```
protected void chkFyzickaOsoba_CheckedChanged(object sender, EventArgs e)
{
    string AddWhere = String.Empty;
    if (chkFyzickaOsoba.Checked == true)
    {
        AddWhere = "IDTypOsoba=2";
    }
    filterSposOsoba.RebuildSqlSelect(String.Empty, String.Empty,
        AddWhere);
}
```

Do parametra *AddWhere* sa zadá, časť Sql príkazu, ktorá nám umožní vybrať zo zoznamu len fyzické osoby. Tento parameter sa zadá do tela komponenty *PxFilterView* cez metódu *RebuildSqlSelect()*, pomocou ktorej sa vygeneruje nový Sql príkaz.

Pri odškrtnutí komponenty *CheckBox*, by sa znova zavolala metóda „chkFyzickaOsoba_CheckedChanged“, kde parameter *AddWhere* by bol prázdny, a tak by sa nám vylistoval zoznam zo všetkými osobami.

Tu si treba ešte všimnúť komponentu *PxFilterView*, ktorá má nastavenú property *TableEnd* na *false*, čo nám umožňuje vložiť komponentu *CheckBox* do tabuľkového riadku a tak zakončiť tabuľku tagom „</table>“ a zarovnať ju s komponentou *PxFilterView*. Výsledný obrázok komponenty *PxFilterView* je na obrázku dole:

Juridical Form: 
Region: 
District: 
Municipality: 
Specialization: 
☐ Zobrazit len fyzickú osobu

Pokiaľ by sme chceli monitorovať, výstupný SQL príkaz, ktorý sa generuje v tele komponenty PxFilterView pri jej zmene, môžeme na to použiť udalosť (event) "MonitorSQLSelect" komponenty PxFilterView.

Použitie komponenty PxFilterView môžete nájsť v príklade s názvom

510_CD_Titul_Px_Fram_Ajax, v každom inštaláčnom balíčku Px Frameworku aj s príkladmi.

1.18. PxUploader – komponenta na nahrávanie binárnych a textových súborov na server

Komponenta **PxUploader** zabezpečuje nahranie binárnych alebo textových súborov z klientského prehliadača na server. **PxUploader** pracuje s prehliadačmi ako Windows Internet Explorer, Firefox a pod. Na správnu funkciu komponenty PxUploader musíte mať vo svojom webovom prehliadači nainštalovaný Media Flash Player a povolený JavaScript.

Pokiaľ chceme pracovať s komponentou PxUploader, musíme si ju v súbore *.aspx zadať nasledovne:

```
<Prx:PxUploader ID="Uploader" runat="server" />
```

Ak chceme, aby komponenta PxUploader bola viditeľná v aspx súboroch, musíme si v hlavičke súboru zadať "Prx" prefix s prepojeným na assembly komponenty PxUploader. Toto vykonáme nasledovne:

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

Keď máme toto vykonané, môžeme v súbore *.aspx.cs s komponentou PxUploader pracovať nasledovne. Najprv si do komponenty PxUploader zadáme cestu do adresára, kde chceme odkladať súbory, cez property *UploadDir*. Ďalej môžeme do property *IdUploadFile* zadať Id súboru, ktoré bude pripojené k menu súboru ako prefix, pre lepšiu jedinečnú identifikáciu súboru. Cez property *MultiFiles*, môžeme nastaviť, či pri výbere sa nám bude dať vybrať jeden alebo viac súborov, ktoré budú nahrávané na server. Viac príklad dole:

```

Uploader.UploadDir = @"uploads\data";
Uploader.IdUploadFile = 7777;
Uploader.MultiFiles = false;

```

Cez property *FileDescription* a *FileExt*, môžeme ovládať zadanie filtra vo formulári OpenFileDialog, kde sa vykonáva výber súborov, podľa prípony súboru.

```
Uploader.FileDescription = "Image Files";
```

```
Uploader.FileExt = "*.jpg;*.png;*.gif;*.bmp;*.jpeg;*.zip;*.rar";
```

Property *SizeLimit* slúži na zadanie maximálnej veľkosti súboru v bajtoch, ktorý môžete nahráť na server. Komponenta PxUploader umožňuje nahráť súbor na server až do veľkosti 2GB. Viac príklad dole, ktorý s aktuálnym prednastaveným nám umožňuje nahráť súbor na server o maximálnej veľkosti 10MB:

```
Uploader.SizeLimit = 10485760;
```

Defaultná veľkosť súboru ktorý je možno nahráť na server, je v IIS nastavený na 4MB. Toto sa dá zmeniť cez nastavenie v súbore web.config, v časti `<system.web>` v property *maxRequestLength*. Viac príklad dole:

```
<system.web>
  <httpRuntime executionTimeout="110" maxRequestLength="2097151"
    requestLengthDiskThreshold="80" useFullyQualifiedRedirectUrl="false"
    minFreeThreads="8" minLocalRequestFreeThreads="4" appRequestQueueLimit="5000"
    enableKernelOutputCache="true" enableVersionHeader="true"
    requireRootedSaveAsPath="true" enable="true" shutdownTimeout="90"
    delayNotificationTimeout="5" waitChangeNotification="0" maxWaitChangeNotification="0"
    enableHeaderChecking="true" sendCacheControlHeader="true"
    apartmentThreading="false"/>
</system.web>
```

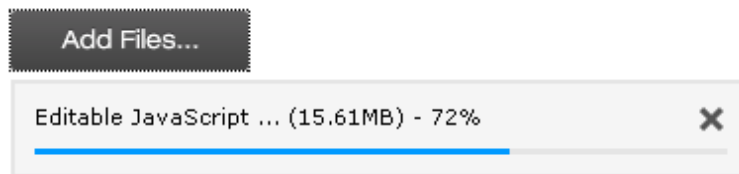
Pokiaľ by sme chceli na server nahrávať súbory do veľkosti 2GB, na IIS verzii 7 musíme ešte v súbore web.config nastaviť property *maxAllowedContentLength* v sekcii `<system.webServer>`. Samozrejme tiež nezabudnúť upraviť property *SizeLimit*, a nastaviť tam veľkosť 2GB. Viac príklad dole:

```
<system.webServer>
  <security>
    <requestFiltering>
      <requestLimits maxAllowedContentLength="2097151000" />
    </requestFiltering>
  </security>
</system.webServer>
```

Cez property *ButtonText*, môžete zmeniť názov tlačítka komponenty PxUploader. Viac príklad dole:

```
Uploader.ButtonText = "Add Files...";
```

Obrázok komponenty PxUploader, pri nahrávaní súboru na server, vyzerá nasledovne, viac obrázok dole:



Udalosti komponenty PxUploader:

Komponenta PxUploader generuje dve udalosti (eventy). Sú to udalosti CompleteUpload a AllCompleteUpload. Udalosť CompleteUpload je zavolaná po kompletnom nahraní súboru na server do adresára, ktorý bol stanovený v property *UploadDir*. V tejto metóde ktorá je zavolaná, sú aj odovzdané premenné, ako názov súboru a cesta k súboru kde bol súbor uložený. Udalosť AllCompleteUpload je zavolaná po udalosti CompleteUpload, spravidla keď boli už všetky súbory prenesené na server. Definíciu jednotlivých udalostí, a ich pripojenie na metódy, nájdete popísané, viac príklad dole:

```
protected void Page_Load(object sender, EventArgs e)
{
    Uploader.IdUploadFile = 7777;
    Uploader.UploadDir = @"uploads\data";
    Uploader.ButtonText = "Add Files...";
    Uploader.CompleteUpload += new
    PxUploader.CompleteUploadEventHandler(Uploader_CompleteUpload);
    Uploader.AllCompleteUpload += new
    PxUploader.AllCompleteUploadEventHandler(Uploader_AllCompleteUpload);
}

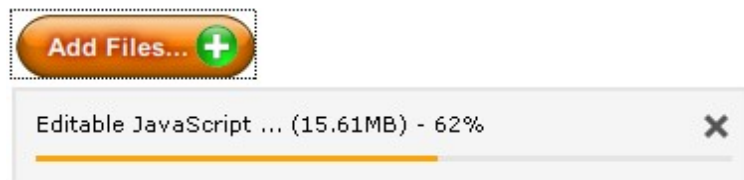
void Uploader_AllCompleteUpload(object sender, EventArgs e)
{
    string a = "";
}

void Uploader_CompleteUpload(object sender, string FileName, string Path, EventArgs e)
{
    string a = "";
}
```

Pokiaľ by sme chceli zmeniť farbu a tvar tlačítka komponenty PxUploader, môžeme tak urobiť cez property *ButtonImageUrl*, do ktorej priradíme názov obrázku tlačítka aj s cestou kde sa nachádza. Viac príklad dole:

```
Uploader.ButtonImageUrl = "../images/AddFiles.png";
Uploader.ButtonWidth = 122;
Uploader.ButtonHeight = 34;
```

Výsledný obrázok komponenty PxUploader, pri nahrávaní súboru na server, so zmeneným tlačítkom, bude vyzeráť nasledovne, viac obrázok dole:



Farbu progres baru môžeme meniť cez property *ProgressBarColor*, zadaným buď textovej hodnoty farby, napr. „red“, alebo zadaným číselnej hodnoty napr. „#FF9933 “. Viac príklad dole:

```
Uploader.ProgressBarColor = "orange";
```

1.19. PxDBPopUp – komponenta pre načítanie dát z komponenty PxWebQuery a ich zobrazenie v prehľadnej tabuľke, výber hodnoty v tabuľke a jej následne zapísanie do editačného prvku, napr. PxEdit, TextBox, PxComboBox apod.

Komponenta PxDBPopUp nám slúži na načítanie obsahu dát z komponenty PxWebQuery, a ich zobrazenie v prehľadnom PopUp okienku. V tomto PopUp okienku si môžeme zvoliť konkrétny riadok tabuľky, a prenesenie jeho hodnoty, konkrétneho stĺpca po stlačení tlačítka Ok do editačného prvku, ako je napr. PxEdit, TextBox, ComboBox a pod. PopUp okienko, nie je riešené cez vyskakovacie PopUp okienko JavaScriptu ale je to jednoduchý tag DIV do ktorého sa premieta obsah okienka, takže táto komponenta je značne rýchla. To ako vyzerá výsledné PopUp okienko, uvidíte, viac obrázkov dole:



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

```
<Prx:PxDBPopUp ID="dbpKraj" runat="server" />
```

Takto zadefinovaná komponenta *PxDBPopUp* sa nám vo výstupnom formulári zobrazí ako tlačítko, na ktoré keď klikneme, otvorí sa nám PopUp okienko. Obrázok tlačítka komponenty *PxDBPopUp*, môžete vidieť, viac obrázkov dole:



Keď máme toto vykonané, môžeme v súbore *.aspx.cs s komponentou *PxDBPopUp* pracovať nasledovne. Do property Caption, komponenty *PxDBPopUp* zadáme názov titulu PopUp okienka a do property PxDataSource sa priradí konkrétna komponenta PxWebQuery, z ktorej sú načítané dáta do prehľadnej tabuľky komponenty *PxDBPopUp*. Property VisibleFields slúži na redukciu počtu zobrazených stĺpcov v okienku a samozrejme, poradie zapísaných stĺpcov sa premietne do poradia stĺpcov, ako sú zobrazené v komponente *PxDBPopUp*. Viac, výpis zdrojového kódu dole:

```
dbpKraj.Caption = "Zvoľte kraj";  
dbpKraj.VisibleFields = "IDKR;NAZKR";  
dbpKraj.PxDataSource = wquKraj;
```

Veľkosť PopUp okienka môžeme ovládať cez property Width a Height, viac príklad dole:

```
dbpKraj.Width = 300;  
dbpKraj.Height = 200;
```

Pozíciu PopUp okienka vo formulári si môžete nastaviť cez property Top a Left, viac príklad dole:

```
dbpKraj.Top = 200;  
dbpKraj.Left = 200;
```

Komponentu *PxDBPopUp*, vlastne jej tlačítko, obyčajne umiestňujeme za komponentov PxEdit, TextBox a pod. Pri výbere konkrétneho riadku, sa nám obsah zvoleného stĺpca, pri stlačení tlačítka „Ok“ prenesie do asociovaného editačného prvku, ktorý môže byť napríklad komponenta PxEdit, alebo komponenta PxComboBox.

Viac obrázok dole:



Konkrétny editačný prvok PxEdit, TextBox a pod. môžeme prepojiť z komponentou *PxDBPopUp*, nasledujúcim spôsobom. Cez property OutKomponentName, kde zadávame hodnotu vlastnosti ID editačného prvku komponent ako PxEdit, TextBox a pod. Do property OutFieldName zadávame meno, názov stĺpca z ktorého sa nám vyberie vyselektovaná hodnota daného riadku.

```
dbpKraj.OutKomponentName = edtKraj;  
dbpKraj.OutFieldName = "NAZKR";
```

Tu v ďalšom nájdete výpis kompletného zdrojového kódu pre súbor *.aspx.

```
<Prx:PxWebQuery ID="wquKraj" runat="server" Value="wquKraj"/>  
  
<Prx:PxEdit ID="edtKraj" runat="server" ></Prx:PxEdit>  
<Prx:PxDbPopUp ID="dbpKraj" runat="server" />
```

A tu je ďalej výpis zdrojového kódu pre súbor *.aspx.cs, z kompletným načítaným dát do komponenty PxWebQuery v metóde Page_Load() daného formulára.

```
protected void Page_Load(object sender, EventArgs e)  
{  
    if ((wquKraj.Active == false))  
    {  
        wquKraj.ConnectionString = AppConst.ConnectionString;  
        wquKraj.SQLSelect = "select IDKR, NAZKR, SKR from KRAJ order by NAZKR ";  
  
        wquKraj.Open();  
  
        wquKraj.Columns["IDKR"].Caption = "Id";  
        wquKraj.Columns["NAZKR"].Caption = "Názov Kraja";  
  
    }  
    else  
    {  
    }  
    dbpKraj.Caption = "Zvoľte kraj";  
    dbpKraj.VisibleFields = "IDKR;NAZKR";  
    dbpKraj.PxDataSource = wquKraj;  
    dbpKraj.Width = 300;  
    dbpKraj.Height = 200;  
  
    dbpKraj.OutKomponentName = edtKraj;  
    dbpKraj.OutFieldName = "NAZKR";  
}
```

Vzhľadom na to, že komponenta *PxDbPopUp* je postavená na JavaScripte, je ju možné použiť len na načítanie maximálneho počtu riadkov, okolo 1000. Ak by ste chceli načítať tabuľky, z väčším počtom riadkov, v takomto prípade je lepšie poďeliť zobrazenie väčšej tabuľky do skupín a potom si cez komponentu *DropDownList* vyberať dáta za danú skupinu.

Na to aby sme mohli vyberať dáta podľa skupiny, musíme zadať field, stĺpec podľa ktorého budeme filtrovať dáta. Tu dole uvádzame príklad za tabuľku obcí, kde počet obcí prekračuje kritický limit komponenty *PxDbPopUp*. Takže obce budú filtrované podľa okresu. Viac zdrojový kód dole, kde do property *SelectFieldName*, zadáme názov stĺpca okresu a do property *CaptionSelectField*, zadáme popisok *DropDownListu*, v ktorom si budeme vyberať daný okres.

```

dbpObec.Caption = "Zvoľte Obec";
dbpObec.VisibleFields = "KODOB;NAZOBCE;NAZOKS";

dbpObec.PxDataSource = wquUTJF;
dbpObec.Width = 500;
dbpObec.Height = 350;
dbpObec.OutKomponentName = edtObec;
dbpObec.OutFieldName = "NAZOBCE";

dbpObec.SelectFieldName = "NAZOKS";
dbpObec.CaptionSelectField = "Zvoľte Okres:";

```

Viac obrázok dole:

...	KODOB	Názov Obce	Názov okresu
	508454	Badín	Banská Bystrica
	508471	Baláže	Banská Bystrica
	508438	Banská Bystrica	Banská Bystrica
	508977	Sebedín - Bečov	Banská Bystrica
	508675	Brusno	Banská Bystrica
	508519	Čerín	Banská Bystrica
	508519	Čerín	Banská Bystrica
	508543	Dolná Mičaná	Banská Bystrica
	508551	Dolný Harmanec	Banská Bystrica
	508560	Donovaly	Banská Bystrica
	508586	Dúbravica	Banská Bystrica
	508594	Harmanec	Banská Bystrica
	508616	Hiadef	Banská Bystrica

Cez DropDownList „Zvoľte Okres“, si vyfiltrujete obce za daný okres a potom kliknutím na daný riadok a potvrdeným tlačítko „Ok“ vyberiete danú obec.

Niekedy nám komponenta *PxDBPopUp* nevie priradiť hodnotu do editačného prvku, aj keď sme do property *OutKomponentName* zadali správne ID editačnej komponenty. Vtedy je vhodné otvoriť si výslednú stránku a pozrieť si jej zdrojový kód, a tam nájsť skutočné meno (Name) komponenty editačného prvku. Obzvlášť pokiaľ sa komponenta nachádza v *UpdatePanely*, komponenta *PxDBPopUp* si ho nevie zistiť, preto je potrebné pozrieť sa do zdrojového kódu, aké meno bolo pridelené komponente behovým prostredím ASP.NET. Potom toto meno zadáme do property *OutKomponentName*, ako reťazec. Viac príklad dole:

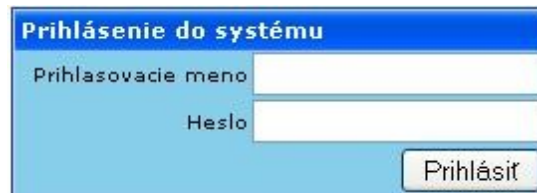
```

dbpKraj.OutKomponentName = "ctl00_cphContent_edtNazKraj";
dbpKraj.OutFieldName = "NAZKR";

```

1.20. PxLogin – komponenta pre autorizáciu a prihlásenie sa do aplikácie

Komponenta PxLogin slúži pre autorizáciu a prihlásenie sa do aplikácie. Komponenta obsahuje dva TextBoxy do ktorých sa zadáva meno a heslo užívateľ, a tlačítko pre potvrdenie a prihlásenie sa do aplikácie. Viac obrázkov dole:



Vytvoríme si súbor Login.aspx, a do neho zapíšeme nasledujúcu definíciu komponenty PxLogin:

Súbor Login.aspx:

```
<Prx:PxLogin ID="lgLogin" LoginFormName="Login.aspx"
  FirstFormAfterLogin="Adresar.aspx" runat="server" />
```

Ako je vidieť z definície komponenty PxLogin, pri jej definícii sa definujú aj dve property a to sú *LoginFormName* a *FirstFormAfterLogin*. Komponentu PxLogin vložíme do každého formulára, a keď tak urobíme, komponenta PxLogin nám bude robiť strážnika, a zisťovať či sme prihlásený v systéme. Ak nie sme prihlásený v systéme, tak komponenta PxLogin nás presmeruje z pôvodného formulára do formulára ktorý je zadaný v property *LoginFormName*, v tomto prípade to je formulár Login.aspx. Ak sme prihlásený v systéme, komponenta PxLogin to vie zistiť zo svojej štruktúry, ostávame v pôvodnom formulári.

Keď do jednotlivých TextBoxov zadáme meno a heslo, potvrdíme to tlačítkom „Prihlásiť“, tak nám komponenta PxLogin zavolá udalosť „ValidateLogin“. Tejto udalosti komponenty PxLogin v tele metódy Page_Load, priradíme procedúru kde sa bude táto udalosť spracovávať. Viac príklad dole:

Súbor Login.aspx.cs:

```
protected void Page_Load(object sender, EventArgs e)
{
    ...
    lgLogin.ValidateLogin += new
        ValidateLoginEventHandler(lgLogin_ValidateLogin);
    ...
}
```

Metóda *lgLogin_ValidateLogin* obsahuje vstupné parametre, a to sú meno a heslo, ktoré bolo zadané vo formulári komponenty PxLogin. Okrem týchto vstupných parametrov obsahuje výstupné parametre a to sú *AcceptLogin* a *ErrorMsg*.

V skratke povedané v metóde *lgLogin_ValidateLogin* prebieha celá autorizácia, a ak meno a heslo je správne, tak do parametra *AcceptLogin* zapíšeme true, ak nie tak false. V prípade ak sme

zapísali false, tak do parametra *ErrorMsg* zapíšeme aj chybovú správu. V prípade ak meno a heslo vyhovelo autorizácii a sú správne a do parametra *AcceptLogin* zapíšeme true, komponenta *PxLogin* nás automaticky presmeruje do formulára, ktorého názov je zapísaný v property

FirstFormAfterLogin.

V tomto prípade je to formulár *Adresar.aspx*. Viac príklad dole:

```
void lgLogin_ValidateLogin(object sender, EventArgs e, string LoginName,
string Passwd, ref bool AcceptLogin, ref string ErrorMsg)
{
    string GroupUsers = String.Empty;
    Int32 IdUser = -1;
    AcceptLogin = IsValidLogin(LoginName, Passwd, ref GroupUsers,
        ref IdUser);
    if (AcceptLogin == true)
    {
        lgLogin.GroupUsers = GroupUsers;
        lgLogin.IDUser = IdUser.ToString();
        Session["GroupUsers"] = GroupUsers.RemoveDiacritics();
    }
    else
    {
        ErrorMsg = "Nezadali ste správne meno alebo heslo !!!";
        lgLogin.GroupUsers = String.Empty;
        lgLogin.IDUser = "-1";
        Session["GroupUsers"] = null;
    }
}
```

Pomocou komponenty *PxLogin* vieme vyvolať presmerovanie z prihlasovacej stránky na stránku, ktorá je určená v property *FirstFormAfterLogin*. Samozrejme, okrem tohto presmerovania komponenta *PxLogin* má ešte inú funkciu, pomocou ktorej môžeme presmerovať beh aplikácia na inú stránku po úspešnom prihlásení. Je to možné pomocou zadania parametra „*redirect_page*“ do url adresy prihlasovacieho formulára. Komponenta *PxLogin* pri svojom načítaní preskúma url adresu, a ak tam nájde parameter „*redirect_page*“, presmeruje beh aplikácie na stránku uvedenú za týmto parametrom, samozrejme až po úspešnom prihlásení. Dole je uvedený vzor url adresy. Všetky ostatné parametre, okrem parametra „*redirect_page*“ sa prekopírujú do novej, presmerovanej url adresy.

Url adresa pomocou ktorej sa zavolá prihlasovací formulár:

http://localhost:3167/Output/login.aspx?redirect_page=posting.aspx&mode=reply&t=2

Url adresa, ktorá je vygenerovaná z komponenty *PxLogin*, po úspešnom prihlásení:

<http://localhost:3167/Output/posting.aspx?mode=reply&t=2>

1.21. PxSpeedButton – varianta tlačítka s obrázkom

Komponenta PxSpeedButton je kombinácia tlačítka s obrázkom a textom. V komponente je možné meniť polohu obrázku, poprípade použiť tlačítko na zaznamenanie stavu (Up / Down).

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

```
<Prx:PxSpeedButton ID="sbtnFind" runat="server"
ImageUrl="img/btnFind.png" Caption="Hľadať"
OnClick="sbtnFind_Click" />
```

Takto zadefinovaná komponenta *PxSpeedButton* sa nám vo výstupnom formulári zobrazí ako tlačítko. Obrázok tlačítka komponenty *PxSpeedButton*, môžete vidieť, viac obrázkov dole:



Ako vidieť v hore uvedenej definícii, tak cez vlastnosť *ImageUrl* sme si zadefinovali relatívnu url adresu k obrázku. Čo sa týka obrázku, tento obrázok je vhodné zadávať vo formáte gif alebo png, s nastavenými transparentnými okrajmi obrázku. To ako urobiť transparentný obrázok pomocou Gimpu, môžete nájsť na tejto adrese (<http://www.proximasoft.sk/clanky/PxSpeedButton-Gimp-Vytvorenie-transparentneho-obrazku.html>).

Vo vlastnosti *Caption* si zas zadefinujeme popisok, text tlačítka. V udalosti *OnClick*, si definujeme názov metódy, ktorá bude zavolaná po stlačení tlačítka. Ďalšou vlastnosťou, ktorú má komponenta PxSpeedButton je vlastnosť *Font*, viac definícia dole v súbore *.aspx:

```
<Prx:PxSpeedButton ID="sbtnFind" runat="server"
Font-Name="Arial, Sans Serif" Font-Size="Medium"
Font-Bold="true" ImageUrl="img/btnFind.png" Caption="Hľadať"
/>
```

Cez vlastnosť *Font-Name* môžeme zadať názov fontu, ktorý sa použije pri vypísaní textu tlačítka. Cez vlastnosť *Font-Size* zas zadáme veľkosť fontu a cez *Font-Bold*, či má byť zobrazený text vypísaný hrubým textom.

```
<Prx:PxSpeedButton ID="sbtnFind" runat="server"
Layout="lbTop" ImageUrl="img/btnFind.png" Caption="Hľadať" />
```

Cez vlastnosť *Layout*, môžeme nastaviť pozíciu obrázka, buď vľavo alebo vpravo, poprípade hore alebo dole. V hore uvedenej definícii sme umiestnili obrázok hore nad text, tým že sme vlastnosti *Layout* priradili hodnotu *lbTop*.

Viac dole na obrázku tlačítka PxSpeedButton:



Ďalšiu vlastnosť, ktorú má komponenta *PxSpeedButton*, je vlastnosť *AllowUp*. Keď túto vlastnosť *AllowUp* nastavíme na hodnotu *false*, tlačítko komponenty začne mať dva stavy. Keď tlačítko stlačíme ostane nám v pozícii zatlačenia, teda dole. Ak naň klikneme ešte raz, tlačítko sa vráti do pôvodnej polohy. Cez vlastnosť *Down* môžeme nastaviť v akej východzej polohe sa má tlačítko zobrazit', či hore alebo dole, či bude zatlačené alebo nie. Viac definícia dole:

```
<Prx:PxSpeedButton ID="sbtnFind" runat="server"
  ImageUrl="img/btnFind.png" Caption="Hľadať"
  AllowUp="false" Down="true" />
<Prx:PxSpeedButton ID="sbtnStorno" runat="server"
  ImageUrl="img/btnStorno.png" Caption="Zavrieť"
  AllowUp="false" Down="false" />
```

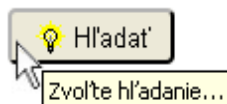
Obrázok ku hore uvedenej definícii dvoch tlačítok, s jedným tlačítkom stlačeným a druhým nie.



Bublinovú nápovedu u komponenty *PxSpeedButton*, môžeme vytvoriť, cez zadanie textu do vlastnosti *ToolTip*, viac definícia obrázok dole:

```
<Prx:PxSpeedButton ID="sbtnFind" runat="server"
  ToolTip="Zvoľte hľadanie..." ImageUrl="img/btnFind.png"
  Caption="Hľadať" />
```

Tlačítko *PxSpeedButton* zobrazené s bublinovou nápovedou:



Pokiaľ by sme chceli ujednotiť dĺžku tlačítka, môžeme tak urobiť cez property *Width*. Viac definícia dĺžky tlačítka dole:

```
<Prx:PxSpeedButton ID="sbtnOk" runat="server"
  Width="80px" ImageUrl="img/btnOK.png" Caption="Ok" />
```

Samozrejme nemali by sme zabudnúť na udalosť *OnClick* komponenty *PxSpeedButton*. Po stlačení tlačítka sa nám zavolá procedúra ktorá bola daná v property *OnClick* komponenty *PxSpeedButton*.

V súbore *.aspx sa udalosť *OnClick* v komponente *PxSpeedButton* definuje nasledovne:

```
<Prx:PxSpeedButton ID="sbtnFind" runat="server"
  ImageUrl="img/btnFind.png" Caption="Hľadať"
  OnClick="sbtnFind_Click" />
```

V súbore *.aspx.cs sa metóda udalosti *OnClick* komponenty *PxSpeedButton* definuje nasledovne:

```
protected void sbtnFind_Click(object sender, EventArgs e)
{
    sbtnFind.Down = true;
}
```

Pokiaľ by sme chceli pre tlačítko použiť vlastný štýl, môžeme to urobiť cez vlastnosti *CssClass* a *CssClassDown*. Trieda *CssClassDown* slúži na definíciu štýlu tlačítka v stave stlačenia, čiže *Down = true*.

1.22 PxMenu – komponenta pre zobrazenie dynamického menu

Komponenta *PxMenu* je komponentou, ktorá umožňuje vytvorenie dynamického vertikálneho menu, a to buď vytvoreným programovo, alebo načítaním jeho štruktúry zo súboru.

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

```
<Prx:PxMenu ID="Menu1" runat="server" />
```

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

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

Keď máme toto vykonané, môžeme v súbore *.aspx.cs s komponentou *PxMenu* pracovať nasledovne. Najprv si do komponenty zadáme jednotlivé položky menu, cez príkaz *PxMenu.Add()*. Viac príklad dole:

```
if ((!this.IsPostBack) && (Menu1.Count==0))
{
    Menu1.Add("N1", "Hlavná ponuka", "~/Produkt.aspx", "");
    Menu1.Add("N2", "Produkt", "~/Produkt.aspx", "N1", "images/insert.gif");
    Menu1.Add("N3", "Výrobca", "~/Vyrobc.aspx", "N1", "images/edit.gif");
    Menu1.Add("N4", "Klient", "~/Klient.aspx", "N1", "images/info.png");
    Menu1.Add("N5", "Číselník", "~/Kategoria.aspx", "");
    Menu1.Add("N6", "Kategória", "~/Kategoria.aspx", "N5");
    Menu1.Add("N7", "Záruka z1", "", "N5");
    Menu1.ModeView = PxControls.PxMenu.enumModeView.mvNormal;
}
```

Pozrime sa bližšie na definíciu príkazu *PxMenu.Add()*, cez ktorý sa pridávajú nové Itemi, nové položky menu.

Popis Parametrov metódy *PxMenu.Add()*:

C# syntax:

```
public void Add(string aName, string aText);
public void Add(string aName, string aText, string aUrlPath);
public void Add(string aName, string aText, string aUrlPath, string
aParentMenu);
public void Add(string aName, string aText, string aUrlPath, string
aParentMenu, string aImageUrl);
```

Popis parametrov:

aName – jedinečné meno položky menu (bez diakritiky)

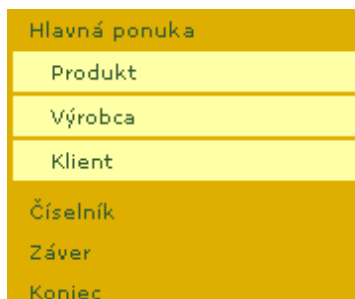
aText – text položky menu, tento text sa zobrazuje v menu pri jeho zobrazení

aUrlPath – webová URL adresa stránky, kde sa presmeruje webová aplikácia, po kliknutí na danú položku menu

aParentMenu – táto položka umožňuje vytváranie podriadeného a nadriadeného menu, každý item menu môže mať aj svojho rodiča (Parenta), potom sa zobrazuje ako submenu svojho parenta

aImageUrl – tu sa zadáva cesta ku ikonke, alebo obrázku položky menu, zobrazuje sa len v režime, kde: `Menu1.ModeView = PxMenu.enumModeView.mvOutlook;`

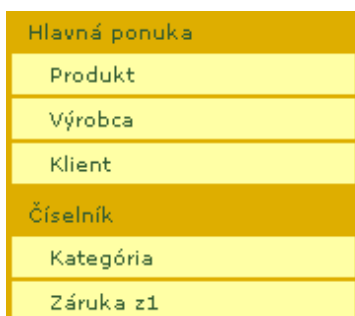
Dole na obrázku je vyobrazená komponenta PxMenu v režime `enumModeView.mvNormal`:



Pokiaľ by sme chceli, aby boli rozbalené všetky submenu, môžeme to urobiť pomocou property *FullExpand* nastavením na hodnotu `true`:

```
Menu1.FullExpand = true;
```

Po aplikácii property *FullExpand*, bude komponenta **PxMenu** v rozbalenom stave, viac obrázok dole:



Pomocou property *SelectedIndex* môžeme nastaviť pozíciu aktívneho menu, teda ktoré submenu má byť otvorené pri spustení aplikácie.

```
Menu1.SelectedIndex = 4;
```

Dĺžku menu môžeme nastaviť cez property *Width*, viac príklad dole:

```
Menu1.Width = Unit.Pixel(180);
```

Komponenta PxMenu podporuje aj štyri módy zobrazenia, a tie sa nastavujú prostredníctvom property *ModeView*:

```
Menu1.ModeView = PxControls.PxMenu.enumModeView.mvNormal;
```

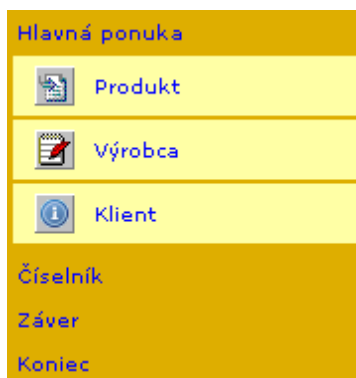
Zoznam jednotlivých hodnôt, ktoré môže nadobúdať výčtový typ(enumeration) *enumModeView* je nasledovný:

```
mvNormal  
mvTree  
mvOutlook  
mvDevelop
```

V móde zobrazenia *mvNormal*, sme si komponentu *PxMenu* už ukázali. V režime zobrazenia *mvTree*, komponenta *PxMenu*, zobrazuje ku svojim položkám aj male navigačné ikonky, viac obrázok dolu:



V móde zobrazenia *mvOutlook*, je možné, každej položke menu prideliť ikonku, obrázok. Cesta k obrázku, sa zadáva prostredníctvom metódy *PxMenu.Add()* a to v parametre *aImageUrl*. Komponenta *PxMenu* v móde zobrazenia *mvOutlook*, vyzerá nasledovne:



```
Menu1.ModeView = PxControls.PxMenu.enumModeView.mvOutlook;
```

V móde zobrazenia *mvOutlook*, môžeme tiež nastaviť aj pozíciu obrázku, či má byť umiestnený vľavo, vpravo, hore alebo dole. Môžete tak vykonať, cez property *ImagePosition*:

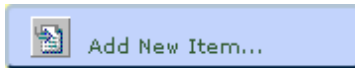
```
Menu1.ImagePosition = PxControls.PxMenu.enumImagePosition.lbTop;
```

Viac obrázok dole, ktorý zobrazuje nastavenie property *ImagePosition* na hodnotu *lbTop*.



Ďalší mód zobrazenia, komponenty PxMenu je *mvDevelop*. Mód zobrazenia *mvDevelop*, umožňuje pokročilé nastavenia menu, a to vytváranie nových položiek menu, ich editáciu, mazanie, posúvanie položiek menu smerom hore, alebo dole, prípadne zmenu rodiča(Parent) pre danú položku menu.

Popis jednotlivých tlačítok menu v móde zobrazenia *mvDevelop*:



- toto tlačítko umožňuje zadať novú položku menu



- toto tlačítko umožňuje editovať už existujúcu položku menu



- toto tlačítko umožňuje zmazať už existujúcu položku menu

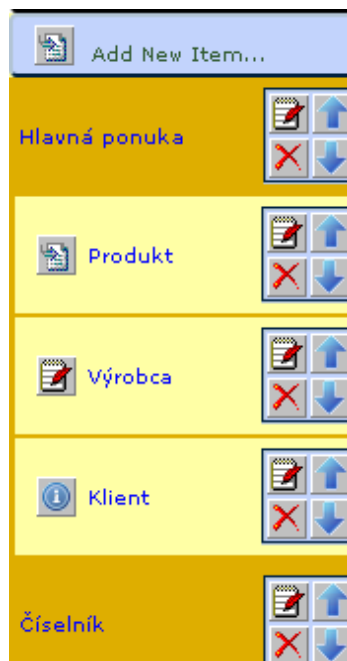


- toto tlačítko umožňuje posúvať existujúcu položku menu smerom hore



- toto tlačítko umožňuje posúvať existujúcu položku menu smerom dole

Viac obrázkov dole komponenty *PxMenu* v móde zobrazenia *mvDevelop*:



Štruktúru menu môžete ukladať do súboru, pomocou príkazu *SaveMenuToFile()*. Toto uloženie štruktúry menu sa vykoná vo formáte JSON.

```
Menu1.SaveMenuToFile(@"e:\aa\Component\IOC\_menu.txt");
```

Podobne môžeme načítať štruktúru menu zo súboru, pomocou metódy *LoadMenuFromFile()*.

```
Menu1.LoadMenuFromFile(@"e:\aa\Component\IOC\_menu.txt");
```

Zneviditeľnenie položky menu môžeme urobiť cez property *Visible*, a jej zneaktívnenie cez property *Enabled*.

```
Menu1.Items[3].Enabled = false;  
Menu1.Items[4].Visible = false;
```

Naštýlovať komponentu PxMenu, môžeme cez súbor Css, keď vypneme property *EnabledInternalStyle = False*. Vzorový css súbor pre komponentu PxMenu môžete stiahnuť na adrese: www.proxima-soft.sk/download/PxMenu-Style.zip

1.23 PxColorComboBox – komponenta pre výber konkrétnej farby z ComboBoxu s farebným rozlíšením

Komponenta *PxColorComboBox* slúži pre výber konkrétnej farby z prehľadného menu v ComboBoxe s farebným rozlíšením.

Komponenta *PxColorComboBox* 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 tu napríklad len telegraficky, zbežne.

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

V súbore **AddCounter.aspx** sa komponenta *PxColorComboBox* definuje nasledovne:

```
<Prx:PxColorComboBox ID="cmbFontColor" runat="server"
AddTableRow="True" Width="250" ></Prx:PxColorComboBox>
```

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

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

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

Súbor AddCounter.aspx.cs:

```
protected void Page_Load(object sender, EventArgs e)
{
    cmbFontColor.PxDataSource = wquCounter;
    cmbFontColor.FieldName = "FontColor";
}
```

Komponenta *PxColorComboBox* má vo svojom tele zadefinované palety farieb. Tieto rôzne palety farieb je možné meniť prostredníctvom property *ColorPallette*. Možné stavy, ktoré môže nadobúdať property *ColorPallette*, sú zadefinované tu dole, viac prehľadná tabuľka:

OrigNamedColors	Kompletný zoznam HTML pomenovaných farieb (napr. "Red")
WebSafeColors	Zoznam ("web safe") bezpečnej palety farieb po 216 farbách. Mená týchto farieb sú reťazce vo formáte HTML v Hex. za zložky RGB (napr. "#FF0000")
OrigNamedColorsOrWebSafeColors	Zlúčenie predchádzajúcich dvoch
ColorsFromDB	Paletu farieb je možné načítať z databázy, z konkrétnej tabuľky, je potrebné na daný stĺpec uvaliť parameter <i>AddParamKey</i> a <i>AddParamWebQuery</i> na komponente <i>PxWebQuery</i>

Nastavenie palety farieb na property *OrigNamedColors*:

Property *ColorPallete* môžeme nastaviť na hodnotu *OrigNamedColors*, toto je najlepšie urobiť v procedúre *Page_Load*, každého formulára, viac výpis dole:

```
protected void Page_Load(object sender, EventArgs e)
{
    cmbFontColor.ColorPalette = PxControls.PxColorComboBox.PxColorComboBoxPalette.OrigNamedColors;
}
```

Keď máme takto nastavenú property *ColorPalette*, po spustení formulára by mala komponenta *PxColorComboBox* vyzeráť nasledovne, viac obrázok dole:

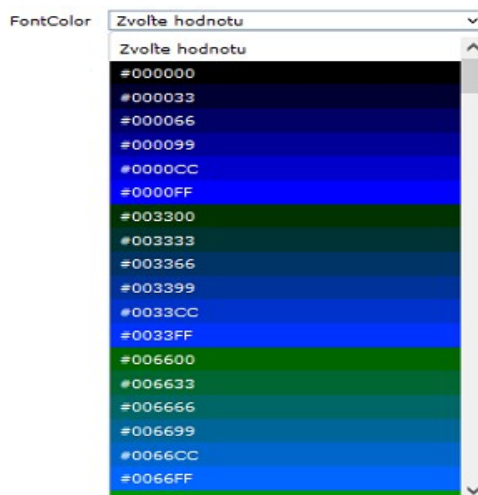


Nastavenie palety farieb na property *WebSafeColors*:

Property *ColorPallete* môžeme nastaviť na hodnotu *WebSafeColors*, toto je najlepšie urobiť v procedúre *Page_Load*, každého formulára, viac výpis dole:

```
protected void Page_Load(object sender, EventArgs e)
{
    cmbFontColor.ColorPalette = PxControls.PxColorComboBox.PxColorComboBoxPalette.WebSafeColors;
}
```

Keď máme takto nastavenú property *ColorPalette*, po spustení formulára by mala komponenta *PxColorComboBox* vyzeráť nasledovne, viac obrázok dole:

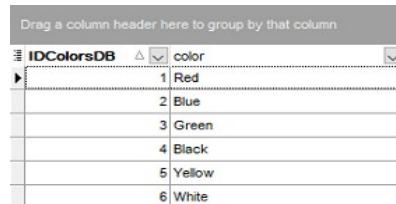


Nastavenie palety farieb na property *OrigNamedColorsOrWebSafeColors*:

Property *ColorPallete* môžeme nastaviť na hodnotu *OrigNamedColorsOrWebSafeColors*, toto je najlepšie urobiť v procedúre *Page_Load*, každého formulára. Táto property je výsledkom zlúčenie dvoch predchádzajúcich palet farieb, teda s originálnym pomenovaným farby a pre Web bezpečných farieb.

Nastavenie palety farieb na property *ColorsFromDB*:

Paletu farieb je možné načítať z databázy, z konkrétnej tabuľky, je potrebné na daný stĺpec uvaliť parameter *AddParamKey* alebo *AddParamWebQuery* na komponente *PxWebQuery*. Najprv si vytvoríme tabuľku v databáze s názvom *ColorsDB* a naplníme do nej hodnoty, viac obrázok dole:



ID	color
1	Red
2	Blue
3	Green
4	Black
5	Yellow
6	White

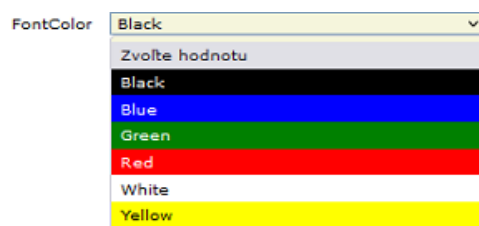
Property *ColorPallete* môžeme nastaviť na hodnotu *ColorsFromDB*, toto je najlepšie urobiť v procedúre *Page_Load*, každého formulára. Potom nastavíme parameter *AddParamKey* na komponente *PxWebQuery* a to konkrétne na *wquCounter*, viac výpis dole:

```
protected void Page_Load(object sender, EventArgs e)
{
    string sSQLSel_wquCounter = "SELECT IDCounter, Counter, FontColor, BackColor, BorderColor FROM counter";
    if ((!IsPostBack) && (wquCounter.Active == false))
    {
        wquCounter.ConnectionString = AppConst.ConnectionString;
        wquCounter.SQLSelect = sSQLSel_wquCounter;

        string sSQLText1 = "select IDColorsDB as aKey, Color as aValue from ColorsDB Order By aValue";
        wquCounter.AddParamKey("FontColor", "aKey", "aValue", "aValue", "aKey", sSQLText1);
        wquCounter.Open();

        wquCounter.Columns["IDCounter"].Caption = "ID";
        wquCounter.Columns["Counter"].Caption = "Counter Value";
        wquCounter.Columns["FontColor"].Caption = "FontColor";
    }
    else
    {
        if ((wquCounter.StateQuery != PxWebQuery.eStateQuery.sqEdit) &&
            (wquCounter.StateQuery != PxWebQuery.eStateQuery.sqInsert))
        {
            wquCounter.SQLSelect = sSQLSel_wquCounter;
            wquCounter.ReOpen();
        }
    }
    cmbFontColor.ColorPalette = PxControls.PxColorComboBox.PxColorComboBoxPalette.ColorsFromDB;
    cmbFontColor.PxDataSource = wquCounter;
    cmbFontColor.FieldName = "FontColor";
}
```

Ak toto všetko máme takto nastavené, komponenta *PxColorComboBox*, by si mala s tabuľky v DB natiahnuť všetky farby a výsledok by mal byť nasledujúci, viac obrázok dole:



Nastavenie funkcie **AutoPostBack** na komponente **PxColorComboBox**:

Funkcia *AutoPostBack* na komponente *PxColorComboBox* zabezpečuje odpálenie formulára, teda odoslanie formulára späť na server v prípade, že sa v komponente *PxColorComboBox* zvolila konkrétna hodnota. Na to aby sa zapla táto funkcia sa musí property *AutoPostBack* zapnúť na *True*. Ďalej sa musí nastaviť event, udalosť *OnSelectedIndexChanged*. Na túto udalosť, teda *OnSelectedIndexChanged* sa pripojí funkcia ktorá sa zavolá pri odpálení formulára.

V súbore **AddCounter.aspx** sa na komponente *PxColorComboBox* nastaví *AutoPostBack* na hodnotu *true* a udalosť *OnSelectedIndexChanged* nasledovne:

```
<Prx:PxColorComboBox ID="cmbFontColor" AddTableRow="True" AutoPostBack="true"
    OnSelectedIndexChanged="SelIndexChange_Click" runat="server"></Prx:PxColorComboBox>
```

Po odpálení formulára sa zavolá procedúra s názvom *SelIndexChange_Click*, kde sa uložia dáta zavolaním procedúry *Post()* na komponente *PxWebQuery*. V tomto prípade komponenta *PxWebQuery* zosníma hodnotu na komponente *PxColorComboBox* a uloží ju do databázy. Procedúrou *Edit()* sa komponenta *PxWebQuery* zase uvedie do stavu *sqEdit*. Je tam aj príkaz na Redirekciu stránky, ktorý slúži na zavolanie celej stránky, pretože v skúšobnom formulári, bol formulár uložený v Updatovacom paneli, ktorý pri odpálení nevedel zabezpečiť farebné prekreslenie. Ak nepoužívate Updatovací panel, môžete tento riadok kľudne vynechať. Viac výpis dole, časť súboru **AddCounter.aspx.cs** a procedúra *SelIndexChange_Click()*:

```
protected void SelIndexChange_Click(object sender, EventArgs e)
{
    wquCounter.Post();
    wquCounter.Edit();
    this.Page.Response.Redirect("~/AddCounter.aspx");
}
```

1.24 PxCounter – komponenta pre zobrazenie počtu prístupov na stránku

Komponenta *PxCounter* je komponentou, ktorá umožňuje zobrazenie počtu prístupov na stránku, jedná sa o malé počítadielko prístupov, spravidla umiestneného na spodnej časti web stránky. Táto komponenta je nevizuálna a na jej plnú funkčnosť ju treba spojiť s HTTP handlerom. Viac obrázkov dole:



V súbore *.ashx.cs sa s komponentou *PxCounter* pracuje nasledovne:

```
PxCounter Counter = new PxCounter();

Counter.MaxNumOfDisplay = 10;    //Počet číslíc na display
Counter.Value = 1;
Counter.FontColor = "#E52B50";   //Farba Fontu
Counter.BackColor = "#C9FFE5";   //Farba podstavy
Counter.BorderColor = "#00308F"; //Farba Rámika
```

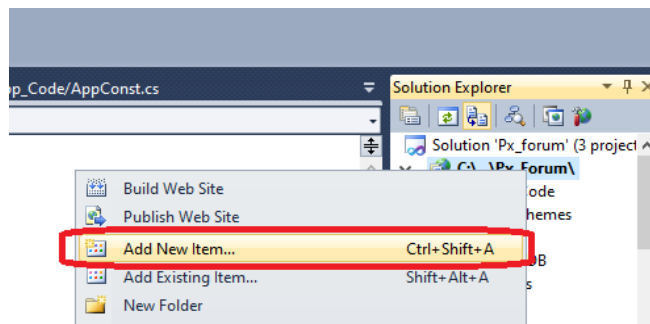
Najprv si komponentu vytvoríme a potom naplníme jej jednotlivé property.

Vlastnosť *MaxNumOfDisplay* slúži na zadefinovanie počtu zobrazených čísiel na displeji, defaultne sa ich zobrazuje 10. Vo vlastnosti *Value* zadávame konkrétnu hodnotu, ktorá sa nám má zobrazovať na displeji. Cez vlastnosti *FontColor*, *BackColor*, *BorderColor* zadávame konkrétnu farbu vo formáte HTML pre Font, Podstavu a Rámik countera. Potom môžeme ešte zadať dĺžku a výšku bunky(*CellWidth*, *CellHeight*) a veľkosť fontu(*FontSize*).

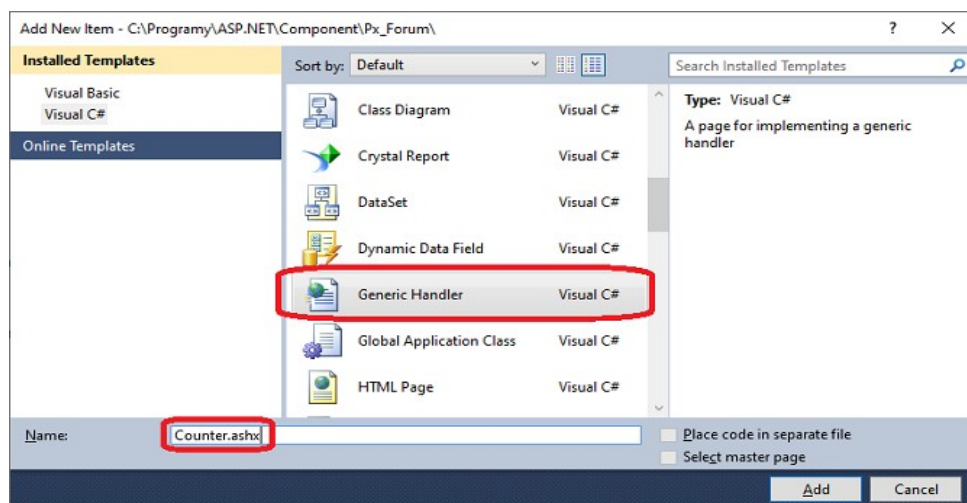
```
CellWidth = 15;
CellHeight = 20;
FontSize = 12;
```

Veľkosť fontu je defaultne nastavená na 12. Ak zmeníme veľkosť fontu, potom musíme dostaviť veľkosť bunky cez vlastnosti *CellWidth*, *CellHeight*. Výsledný *PxCounter* môžeme získať cez funkciu „*GetImageCounter()*“, ktorá nám vráti obrázok vo formáte PNG. Ak chceme komponentu *PxCounter* umiestniť na stránku, prvý krok je vytvoriť konkrétny HTTP Handler do ktorého umiestníme komponentu *PxCounter*.

Pri vytváraní HTTP Handlera si budeme počínať nasledovne. V našom projekte pravým tlačítkom myši klikneme na náš projekt a malo by sa nám zobrazovať nasledujúce menu. Tam zvolíme „Add New Item...“, viac obrázok dole:



Tam si zvolíme „Generic Handler“ a zadáme názov súboru Counter.ashx.



Mal by sa nám otvoriť súbor Counter.ashx a do neho by sme mali vložiť nasledujúci kód.

```
<%@ WebHandler Language="C#" Class="CounterEx" %>

using System;
using System.Web;
using PxControls;

public class CounterEx : IHttpHandler {

    public void ProcessRequest (HttpContext context) {

        PxCounter Counter = new PxCounter();

        Counter.MaxNumOfDisplay = 10;
        Counter.Value = 1;
        Counter.FontColor = "#E52B50";    //Farba Fontu
        Counter.BackColor = "#C9FFE5";    //Farba podstavu
        Counter.BorderColor = "#00308F";  //Farba Rámika

        AppConst.UpdateSQL("update set Counter=Counter+1 where idcounter=1");
        Dictionary<string, string> cnt =
        AppConst.SelectSQL("select Counter,FontColor,BackColor,BorderColor
        where idcounter=1");

        Counter.Value = Convert.ToInt32(cnt["Counter"]);
        Counter.FontColor = cnt["FontColor"];    //Farba Fontu
        Counter.BackColor = cnt["BackColor"];    //Farba podstavu
        Counter.BorderColor = cnt["BorderColor"]; //Farba Rámika
        byte[] buffer = Counter.GetImageCounter();
        context.Response.ContentType = "image/png";
        context.Response.BinaryWrite(buffer);
        context.Response.Flush();
    }

    public bool IsReusable {
        get {
            return false;
        }
    }
}
```

Do webovej stránky, si takto vyrobený PxCounter môžete vložiť cez tag DIV, viac príklad dole:

```
<DIV class="Copyright">
  <A href="http://www.proxima-soft.sk" target="_blank">
    <IMG SRC="http://www.proxima-soft.sk/forum-sk/Counter.ashx"
      alt="counter" border="0" style="margin-top:7px">
  </A>
</DIV>
```

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 hlavičke súboru zadať „Prx“ prefix. Toto vykonáme nasledovne:

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

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

3.1. 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 *ConnectionString*, priradíme *ConnectionString*, a do property *SQLSelect*, priradíme 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.ConnectionString = ""User Id=adr;Password=aa;Data Source=x;";";
        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 nové riadky potrebujete vytvoriť sekvenciu, z ktorej si komponenta *PxWebQuery* načíta jedinečné ID pre svoj primárny kľúč. Názov sekvencie, 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 sekvenciu 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 sekvencia 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 sekvenciu, 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 „%“, funkcionality 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 vyhľadávacieho parameter sa zadáva platná hodnota primárneho kľúča.

3.1.7. Nastavenie pozície kurzora riadku, v komponente PxSuperGrid a PxWebQuery

Nastavenie pozície kurzora v komponente PxSuperGrid môžeme priradeným hodnoty 0..RecordCount-1 do property ItemIndex komponenty PxWebQuery, na ktorú je napojená komponenta PxSuperGrid.

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

3.1.8. 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 vráti objekt DataTable. Objekt DataTable podporuje selektovanie, výber riadkov. Viac príklad dole:

```
//Vypise čísla Id riadkov, ktoré vyhovujú výberovému kritériu  
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 stĺpci „name“, sa v texte nachádza písmeno „a“.

3.1.9. 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 vyzeráť 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 vyvolaní 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 príkazom PxWebQuery.Insert();
AfterInsert	Táto udalosť je vyvolaná po vykonaných príkazoch PxWebQuery.Insert();
BeforeEdit	Táto udalosť je vyvolaná pred vykonaným príkazom PxWebQuery.Edit();
AfterEdit	Táto udalosť je vyvolaná po vykonaných príkazoch PxWebQuery.Edit();
BeforePost	Táto udalosť je vyvolaná pred vykonaným príkazom PxWebQuery.Post();
AfterPost	Táto udalosť je vyvolaná po vykonaných príkazoch PxWebQuery.Post();
BeforeValidPost	Táto udalosť je vyvolaná pri vykonaní verifikácie zadaných položiek, pred samotným uložením dát metódou Post();
BeforeOpen	Táto udalosť je vyvolaná pred vykonaným príkazom PxWebQuery.Open();
AfterOpen	Táto udalosť je vyvolaná po vykonaní príkazu PxWebQuery.Open();
BeforeScroll	Táto udalosť je vyvolaná pred vykonaným zmenou pozície kurzora riadku v komponente PxWebQuery
AfterScroll	Táto udalosť je vyvolaná po vykonaní zmeny pozície kurzora riadku v komponente PxWebQuery
BeforeDelete	Táto udalosť je vyvolaná pred vykonaným príkazom PxWebQuery.Delete();
AfterDelete	Táto udalosť je vyvolaná po vykonaných príkazoch PxWebQuery.Delete();
BeforeCancel	Táto udalosť je vyvolaná pred vykonaným príkazom PxWebQuery.Cancel();
AfterCancel	Táto udalosť je vyvolaná po vykonaných príkazoch PxWebQuery.Cancel();
StateChange	Táto udalosť je vyvolaná pri zmene stavu komponenty PxWebQuery, a to obvyčajne po vykonaní príkazov Insert, Edit, Delete, Post a pod.
AfterCreateField	Táto udalosť je vyvolaná po vytvorení všetkých stĺpcov(fieldov) tabuľky komponenty PxWebQuery. Táto udalosť sa využíva prevažne na vytvorenie nového dynamického stĺpca(fieldu) tabuľky komponenty PxWebQuery.

3.1.10. 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 vytváraní 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	expression – 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 mieste 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)"

3.1.11. 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.ConnectionString = "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 nezádáte žiadnu hodnotu, použije sa defaultna hodnota

120 sekúnd, teda ku automatickému refrešu bude dochádzať každé dve minúty.

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

3.1.12. 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.

Pridať adresu

ID

Meno * Prekročili ste povolenú dĺžku reťazca, maximálne môžete zadať 50 znakov !!!

Osoby

Adresa

Právna Forma

Dátum založenia * Nezadali ste správny dátum!!!

Pri automatickej validácii, a to kontrole dĺžky reťazca, nemusí validácia fungovať celkom správne. Závisí to od toho, aká kódová stránka je nastavená v databáze. Ak v databáze používate kódovú stránku s UTF8, tak v databáze Oracle znaky s diakritikou zaberajú 2 znaky a niektoré špeciálne znaky až 3 znaky v databáze. Preto musíte nastaviť property CharacterCode komponenty PxWebQuery na hodnotu „UTF8“, aby správne fungovala validácia dĺžky reťazca. V ďalšom ponúkame výpis konštánt property CharacterCode, pre správnu funkcionality validácie dĺžky reťazca. V property CharacterCode je defaultne nastavená hodnota ASCII.

P.č.	Konštanta CharacterCode
1	ASCII
2	UTF8
3	UTF16

4	UTF32
5	UTF7

Pokiaľ máme v databáze nastavenú kódovú stránku s UTF8, mali by sme to nastaviť do property `CharacterCode` pred otvorením komponenty *PxWebQuery*. Viac príklad dole:

```
wquAdresar.CharacterCode = "UTF8";
```

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:

Pridať adresu

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

Pridať adresu

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:

EnumValidationTest	Popis typu validácie
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 validná, 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 validná 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
vtValidToDateTime	pri tomto type je testované, či zadaný výraz je platný dátum
vtValidToLengthString	pri tomto type je testovaná dĺžka reťazca z hodnotou v premennej iMinMaxValue, ak dĺžka reťazca je väčšia ako dĺžka v premennej iMinMaxValue vráti chybovú správu
vtIsUnique	pri tomto type je testované, či zadaná hodnota je unikátna, jedinečná(Unique) pre ten daný stĺpec

Každý typ validácie, má pevne pridelenú chybovú správu. Ak by sme chceli túto chybovú správu 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 sRegVýrazToPSC = @"^\d{5}$|^\d{3}\s\d{2}$";  
wquAdresar.AddParamValidation("psc", "vtValidToRegularExpression", sRegVýrazToPSC, 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 „vtValidToRegularExpression“. 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.

Editovať adresu

3.1.13. 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 vyzeráť nasledovne:

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

3.1.14. 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 ne zadáte jazykovú mutáciu, Px Framework si ju zistí sám z konfigurácie IIS servera, a tú bude používať defaultne pre všetky stránky.

3.1.15. Ďalšie metódy AddParam... komponenty PxWebQuery

3.1.15.A. Metóda AddParamLink komponenty PxWebQuery

Prostredníctvom metódy AddParamLink môžete vytvoriť v komponente PxSuperGrid odkaz, hyperlink na súbor, alebo nejakú inú stránku.

Definícia príkazu AddParamLink:

```
public void AddParamLink(string aFieldName, string aFieldNameText, string aFieldNameFile,
                        string aPathFile, string aExtensionFile)
```

Popis parametrov príkazu AddParamLink:

aFieldName – Meno stĺpca, ktorý sa vytvorí ako nový stĺpec v tabuľke na ktorý sa aplikuje metóda AddParamLink (toto meno stĺpca musí byť jedinečné, v tabuľke nesmie existovať stĺpec s týmto menom). Potom v komponente PxSuperGrid necháme zobraziť stĺpec s týmto menom, a tam sa nám zobrazí odkaz, link nami ďalej zadefinovaný.

aFieldNameText – Meno stĺpca, z ktorého text bude zobrazený ako názov odkazu

aFieldNameFile – Meno stĺpca, kde je definovaný názov súboru alebo stránky, ktorá sa otvorí pri kliknutí na odkaz

aPathFile – Tu sa zadefinuje cesta k súboru, alebo stránke, počíta sa s tým že v stĺpci aFieldNameFile, nie je zadefinovaná úplná cesta ku súboru alebo stránke

aExtensionFile – Tu sa definuje koncovka súboru, extension, pokiaľ nebola definovaná v stĺpci aFieldNameFile

Zo všetkých hore zadefinovaných parametrov sa nám v bunke komponenty PxSuperGrid vytvorí odkaz, link na súbor alebo stránku nami definovanú. V praxi uvalenie parametra AddParamLink na komponentu PxWebQuery vyzerá nasledovne:

```
wquAdresar.AddParamLink("NameLink", "LinkText", "pdffilename", "../files_pdf/", ".pdf");
...
```

Čo prebieha pri vykresľovaní konkrétnej bunky komponenty PxSuperGrid, pri uvalení parametra AddParamLink na komponentu PxWebQuery? Odkaz z vyššie uvedených parametrov, sa nám vytvára, skladá nasledujúcim spôsobom:

```
<a href=[aPathFile][aFieldNameFile][aExtensionFile]> [aFieldNameText]</a>
```

Odkaz v komponente PxSuperGrid, zadefinovaný v metóde AddParamLink, bude vyzeráť nasledovne:

...	ID	Názov	PDF File Name
 	253	C# 2005 programujeme profesionálne	801-122-122
 	254	ASP.NET konfigúrácia a nastavenie	801-122-123
 	255	Začínáme programovať v C#	801-122-124
 PRIDAŤ NOVÝ RIADOK...			
[...] [76] [77] [78] [79] [80] [81] [82] [83] [84] Page 85			

3.1.15.B. Metóda AddParamDynamicField komponenty PxWebQuery

Prostredníctvom metódy AddParamDynamicField, môžeme vytvárať dynamický obsah stĺpcov a zobrazenie dát v bunke komponenty PxSuperGrid. Táto metóda nám vyplní medzeru, a umožňuje nám dynamické definovanie obsahu bunky komponenty PxSuperGrid.

Definícia príkazu AddParamDynamicField:

```
public void AddParamDynamicField(string aFieldName,
    ParamDynamicFieldEventHandler aMethodName)
```

Popis parametrov príkazu AddParamLink:

aFieldName – Meno stĺpca, ktorý bude vytvorený, ako nový stĺpec v tabuľke na ktorý sa aplikuje metóda AddParamDynamicField (toto meno stĺpca musí byť jedinečné, v tabuľke nesmie existovať stĺpec s týmto menom). V komponente PxSuperGrid sa nám v tomto stĺpci, s týmto menom, zobrazí obsah, ktorý bol vygenerovaný metódou aMethodName.

aMethodName – Meno metódy, ktorá sa vždy zavolá pred vykresľovaním obsahu bunky komponenty PxSuperGrid a generuje obsah pre jej vykreslenie. Typ tohto parametra je delegát ParamDynamicFieldEventHandler, ktorý je definovaný v tele komponenty PxWebQuery.

Druhý parameter metódy AddParamDynamicField je delegát, ktorý je definovaný v tele komponenty PxWebQuery. My na tento parameter pripájame metódu nami definovanú, v ktorej si dynamicky definujeme obsah danej konkrétnej bunky, ktorá sa bude vykresľovať v komponente PxSuperGrid. Parameter AddParamDynamicField uvažujeme na komponentu PxWebQuery nasledujúcim spôsobom:

```
wquSposOsoba.AddParamDynamicField("DynamicField", CreateDynamicField_Adresar);
```

Metódu CreateDynamicField_Adresar si definujeme ďalej nasledujúcim spôsobom:

```
protected void CreateDynamicField_Adresar(object sender, DataRow DataRow, bool SelectedRow, ref string Value)
{
    if (DataRow != null)
    {
        Value = "<img src='./foto/' + DataRow[\"fotoname\"] + \".jpg' alt='Foto' width=50 align='left'>";
        Value += "<br /> " + DataRow[\"Name\"].ToString() + ", <br /> " +
            DataRow[\"IDPravnaForma\"].ToString();
    }
    else
        Value = String.Empty;
}
```

Treba si uvedomiť že počet parametrov metódy CreateDynamicField_Adresar , a ich typ je určený definíciou delegáta v tele komponenty PxWebQuery. Tá definícia delegáta vyzerá nasledovne:

```
Delegate void ParamDynamicFieldEventHandler(object sender, DataRow DataRow, bool SelectedRow, ref string Value);
```

Metóda CreateDynamicField_Adresar má ako druhý parameter DataRow, kde je v tomto objekte natiahnutý aktuálny riadok komponenty PxWebQuery, ktorý sa bude vykresľovať. Parameter SelectedRow je typu bool, a slúži nám na identifikáciu aktuálneho riadku, teda riadku na ktorom je nastavený kurzor. Pokiaľ je na riadku nastavený kurzor, tak parameter SelectedRow je „true“, pokiaľ nie, potom je „false“.

Parameter Value je typu „ref“, čo znamená že je to pointer, do ktorého si definujeme návratovú hodnotu, ktorá bude zobrazená v príslušnej bunke komponenty PxSuperGrid.

V metóde CreateDynamicField_Adresar sme si zadefinovali zobrazenie obrázka, a za nim zobrazenie mena a právnej formy osoby. Objekt „DataRow[\"fotoname\"]“ nám vracia meno fotografie, a objekt „DataRow[\"Name\"]“ nám vracia meno osoby. Výsledné zobrazenie komponenty PxSuperGrid nájdete na obrázku dole:

...	ID	DynamicField 1	Dátum založenia	Obec, Okres
   	246	 Peter Dowel, Právnická osoba	17. 11. 2010 12:00:00	Obec: Vysoké Tatry Okres: Poprad Kraj: Prešovský
   	247	 Helena Mociková, Fyzická osoba	23. 11. 2010 12:00:00	Obec: Kuzmice Okres: Topoľčany Kraj: Nitriansky
   	248	 Dana Maeská, Fyzická osoba	4. 11. 2010 12:00:00	Obec: Čab Okres: Nitra Kraj: Nitriansky
   	249	 Juraj Fandly, Fyzická osoba	17. 11. 2010 12:00:00	Obec: Jesenské Okres: Levice Kraj: Nitriansky
 PRIDAŤ NOVÝ RIADOK...				
Page 1 [2]				

3.1.16. Špeciálna funkcia GetValueFromKey komponenty PxWebQuery

Funkcia GetValueFromKey slúži na načítanie hodnoty stĺpca, pokiaľ poznáme hodnotu kľúča, ktorý definuje riadok z výstupnou hodnotou.

Špeciálna funkcia GetValueFromKey komponenty PxWebQuery sa dá použiť vtedy, keď je komponenta PxWebQuery otvorená cez príkaz Open(), a sú v nej natiiahnuté dáta z databázy.

Popis Parametrov funkcie GetValueFromKey:

C# syntax:

```
public string GetValueFromKey(string FieldNameIn, string ValueIn, string FieldNameOut);  
public string GetValueFromKey(string FieldNameIn, Int32 ValueIn, string FieldNameOut);
```

Popis parametrov:

FieldNameIn – Meno stĺpca v tabuľke, podľa ktorého budeme vyhľadávať riadok s návratovou hodnotou

ValueIn – Hodnota, ktorú budeme vyhľadávať v stĺpci **FieldNameIn**, podľa ktorej nájdeme riadok s návratovou hodnotou

FieldNameOut – Hodnota tohto stĺpca z vyhľadaného riadka, je návratovou hodnotou funkcie GetValueFromKey

Return – Pokiaľ sa nenájde riadok, ktorý má hodnotu ValueIn v stĺpci FieldNameIn je vrátená hodnota String.Empty. Pokiaľ sa tento riadok nájde, funkcia vracia hodnotu stĺpca FieldNameOut nájdeného riadka.

V skratke povedané Funkcia GetValueFromKey nám slúži na načítanie ktorejkoľvek hodnoty z tabuľky komponenty PxWebQuery, pokiaľ poznáme kľúč tabuľky a jedinečnú hodnotu tohto kľúča, ktorý definuje riadok s výstupnou hodnotou.

3.1.17. Tri spôsoby použitia komponenty PxWebQuery

Komponenta *PxWebQuery*, umožňuje viacej spôsobov použitia. Jej rozumné použitie dokáže ušetriť výkon počítača a zrychliť beh aplikácie. Ďalej budú uvedené tri spôsoby použitia komponenty *PxWebQuery*, a ich popis vhodnosti pre dané typy situácií.

Spôsoby použitia komponenty PxWebQuery:

1. Komponenta PxWebQuery ostáva otvorená (bez Close();)
2. Komponenta PxWebQuery je uzatvorená cez príkaz Close(); (tzv. mäkké close)
3. Komponenta PxWebQuery je uzatvorená cez príkaz Close(True); (tzv. tvrdé close)

3.1.17.1. Komponenta PxWebQuery ostáva otvorená (bez Close();)

Po načítaní dát do komponenty *PxWebQuery* a ich spracovaní, nie je vykonaný príkaz Close().

Tento spôsob použitia značne zrýchľuje aplikáciu, dáta sú odkladané do pamäte, a prežívajú aj ukončenie behového cyklu stránky. Dáta komponenty *PxWebQuery* pri tomto spôsobe použitia, sa spravidla stratia pri ukončení session daného užívateľa. Tento spôsob použitia komponenty *PxWebQuery* je veľmi vhodný, šetrí databázový server a sieť, urýchľuje výsledné vygenerovanie web stránky.

V príklade dole, je ukázané správne použitie komponenty *PxWebQuery* v tomto prípade.

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

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

Keď máme toto vykonané, môžeme v súbore *.aspx.cs s komponentou *PxWebQuery* pracovať nasledovne. Najprv si do komponenty zadáme *ConnectionString* a potom SQL príkaz. Potom cez príkaz „Open“, načítame do komponenty dáta z databázy, a môžeme ju pripojiť na komponentu *PxSuperGrid*, v ktorej sa zobrazia načítané dáta. Po napojení na komponentu *PxSuperGrid*, nezadáваме príkaz *Close()*, a dáta nám v komponente *PxWebQuery* pretrvávajú až do ukončenia session daného užívateľa.

```
protected void Page_Load(object sender, EventArgs e)
{
    if (wquAdresar.Active==false)
    {
        wquAdresar.ConnectionString = "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();
}
```

Na občasné občerstvenie dát komponenty *PxWebQuery* môžeme použiť automatický refresh, ktorý beží 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
```

3.1.17.2. Komponenta *PxWebQuery* je uzatvorená cez príkaz *Close()*; (tzv. mäkké close)

Po načítaní dát do komponenty *PxWebQuery* a ich spracovaní, je vykonaný príkaz `Close()` alebo `Close(false)`, tzv. mäkké close, uzavretie komponenty a zrušenie jej pamäťového priestoru na konci behového cyklu ASP.NET stránky.

Pri tomto spôsobe použitia, sa po každom behovom cykle stránky dáta komponenty *PxWebQuery* zmažú (vynulujú), a pri každom ďalšom štarte web stránky, sú znova cez príkaz `select...` načítané dáta do komponenty *PxWebQuery*. Tento spôsob použitia, viac zaťažuje databázový server, a spomaľuje výsledné vygenerovanie web stránky. Výhoda tohto spôsobu použitia je tá, že v komponente *PxWebQuery*, sú vždy aktuálne dáta z databázy po každom zobrazení web stránky.

V príklade dole, je ukázané správne použitie komponenty *PxWebQuery* v tomto prípade.

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

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

Keď máme toto vykonané, môžeme v súbore *.aspx.cs s komponentou *PxWebQuery* pracovať nasledovne. Najprv si do komponenty zadáme `ConnectionString` a potom SQL príkaz. Potom cez príkaz „Open“, načítame do komponenty dáta z databázy, a môžeme ju pripojiť na komponentu *PxSuperGrid*, v ktorej sa zobrazia načítané dáta. Po napojení na komponentu *PxSuperGrid*, zadávame príkaz `Close()`, a dáta nám v komponente *PxWebQuery* pretrvávajú až do ukončenia behového cyklu generovanej web stránky.

```
protected void Page_Load(object sender, EventArgs e)
{
    if (wquAdresar.Active==false)
    {
        wquAdresar.ConnectionString = "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();

    wquAdresar.Close();
}
```

Namiesto príkazu `wquAdresar.Close()`, môžete zadať aj príkaz `wquAdresar.Close(False)`, má podobnú účinnosť.

V tomto prípade použitia, sa nedoporučuje používať automatický refresh komponenty *PxWebQuery*.

3.1.17.3. Komponenta *PxWebQuery* je uzatvorená cez príkaz `Close(True)`; (tzv. tvrdé close)

Po načítaní dát do komponenty *PxWebQuery* a ich spracovaní, je vykonaný príkaz `Close(True)`, tzv. tvrdé close. Uzavretie komponenty a zrušenie jej pamäťového priestoru, nastane hneď, ako je zavolaný príkaz `Close(True)`.

Tento spôsob použitia je vhodný pri dynamickom vytvorení komponenty *PxWebQuery*, keď potrebujeme rýchlo načítať špecifické dáta na jedno použitie. Pri dynamickom vytvorení komponenty *PxWebQuery*, je vhodné ukončiť ju cez príkaz `Close(True)`; Dynamické vytvorenie komponenty *PxWebQuery* a jej ukončenie vyzerá nasledovne:

```
PxWebQuery wquAdresar = new PxWebQuery();

wquAdresar.ConnectionString = "User Id=adr;Password=aa;Data Source=xe;";
wquAdresar.SQLSelect = "select idadresar , name, address, create_date from Adresar";
wquAdresar.Open();
.....
.....
wquAdresar.Close(True);
```

V tomto prípade použitia, sa nedoporučuje používať automatický refresh komponenty *PxWebQuery*.

3.1.18. Vytvorenie novej jazykovej mutácie Px Frameworku

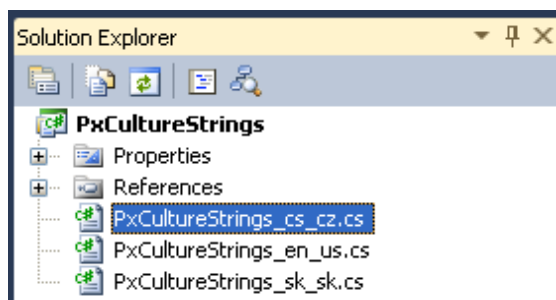
Od verzie 1.21 podporuje Px Framework vytváranie vlastných jazykových mutácií.

Všetky dôležité texty v rôznych jazykových mutáciách sú uložené v assembly PxCultureStrings.dll, ktorá sa nachádza v distribúcii Px Frameworku. Pokiaľ chceme vytvoriť vlastnú jazykovú mutáciu, musíme si stiahnuť celý projekt, cez ktorý si zostavíme výslednu assembly PxCultureStrings.dll.

Celý projekt je uložený v súbore PxCultureStrings.zip a môžete si ho stiahnuť na url :

<http://www.proxima-soft.sk/download/PxCultureStrings.zip>.

Keď si otvoríme projekt PxCultureStrings.csproj v programe MS Visual Studio, mali by sme tam nájsť tri classy s názvom PxCultureStrings_cs_cz.cs, PxCultureStrings_sk_sk.cs a PxCultureStrings_en_us.cs. Class s názvom PxCultureStrings_sk_sk.cs obsahuje slovenskú mutáciu Px Frameworku. Class s názvom PxCultureStrings_en_us.cs zas obsahuje anglickú mutáciu Px Frameworku.



Ak by sme si chceli vytvoriť vlastnú jazykovú mutáciu Px Frameworku, museli by sme vytvoriť novú triedu so špecifickým názvom. Predstavme si, že by sme chceli Px Framework lokalizovať do nemčiny. Potom by sme si museli nájsť v zozname Jazyk-Krajina, kód danej krajiny, ktorá sa používa v triede CultureInfo pod C# a ASP.NET. Tabuľka priložená v projekte PxCultureStrings, zobrazuje výpis všetkých kódov krajín, ktoré používa trieda CultureInfo. Ak by sme si pozreli v tejto tabuľke nemčinu, tak kód pre nemecký jazyk-krajinu bude „de-DE“. Z tohto kódu sa ďalej odvodzuje názov novej triedy, triedy z novou jazykovou mutáciou, ktorá sa pripája do projektu PxCultureStrings.

Výsledný kód, t.j. „de-DE“, sa upraví tak, že všetky veľké písmená sa skonvertujú na malé písmená, a pomlčka sa premení na podtržítka. Výsledok bude vyzeráť takto „de_de“. Ku tomu pripojíme ešte ako prefix nasledujúce slovo „PxCultureStrings_“, a tak získame výsledný názov triedy, a ten je „PxCultureStrings_de_de“.

Podľa vyššie stanoveného algoritmu, môžeme stanoviť všetky názvy tried, podľa kódov jazyk-krajina definovaných v triede CultureInfo.

Teda pre kód „de-DE“, bude výsledný názov triedy vyzeráť takto: „PxCultureStrings_de_de“

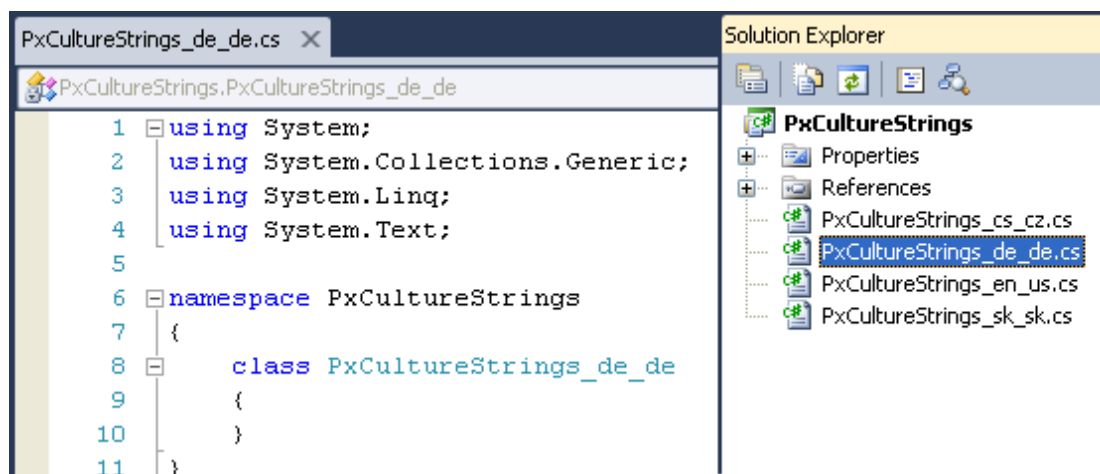
Teda pre kód „fr-FR“, bude výsledný názov triedy vyzeráť takto: „PxCultureStrings_fr_fr“

Teda pre kód „ru-RU“, bude výsledný názov triedy vyzeráť takto: „PxCultureStrings_ru_ru“

Pokiaľ už vieme názov triedy, potom si môžeme v projekte PxCultureStrings, vytvoriť novú triedu s názvom „PxCultureStrings_de_de“.

To urobíme tak, že pravým tlačítkom myši klikneme na projekt PxCultureStrings a potom vyberáme z menu Add->Class...

Otvorí sa nám formulár Add New Item a tam zadáme názov triedy, ktorú chceme vytvoriť. V tomto prípade to je „PxCultureStrings_de_de.cs“. Po potvrdení sa nám nasledujúca trieda vytvorí.



Túto triedu budeme musieť zmeniť na verejnú, t.j. Pridáme tam vlastnosť „`public static`“. Viac príklad dole.

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

namespace PxCultureStrings
{
    public static class PxCultureStrings_de_de
    {
    }
}
```

Potom si z triedy `PxCultureStrings_en_us` prekopírujeme všetky reťazce, do triedy `PxCultureStrings_de_de.cs`. Vyzeráť to bude nasledovne:

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

namespace PxCultureStrings
{
    public static class PxCultureStrings_de_de
    {

        //*****Language Strings For PxComboBox and PxFlyComboBox****
        public static string sChooseValue = "Choose value";
        //***END***Language Strings For PxComboBox and PxFlyComboBox****

        //*****Language Strings For PxJSDatePicker ****
    }
}
```

```

public static string sDateFormat = "MM/dd/yyyy";
//***END***Language Strings For PxJSDatePicker ****

//*****Language Strings For PxValidation ****
public static string sMsgEmptyString = "???";
public static string sValidToDateTime = "You have not entered the correct
date !!!";
public static string sIsNotNull = "You did not value !!!";
public static string sValidToDate = "You have not entered the correct
date !!!";
public static string sValidToEmail = "You have not entered the correct
email address !!!";
public static string sValidToLengthString = "You have exceeded the allowed
length of string, you can specify the maximum {0} characters !!!";

...

```

Všetky texty, ktoré sú v úvodzovkách, prekladáme do nemeckého jazyka. Všetko ostatné, ostáva nemenné, nemeníme názvy premenných, ich typ a ani ostatné vlastnosti.

Keď máme dokončený preklad všetkých textov, projekt zkompilujeme a výslednú knižnicu PxCultureStrings.dll priložíme ku distribúcii Px Frameworku, prepíšeme pôvodnú knižnicu PxCultureStrings.

Potom pri štarte aplikácie a v každom formulári, v procedúre „Page_Load“ musíte zadať nasledujúci zdrojový kód, aby sa načítala nemecká jazyková mutácia Px Frameworku, viac výpis zdrojového kódu dole:

```

protected void Page_Load(object sender, EventArgs e)
{
    PxCulture.CurrentCulture = CultureInfo.CreateSpecificCulture("de-DE");
    ...
}

```

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. Ak neexistuje defaultná verzia jazykovej mutácie Px Frameworku, potom Px Framework použije ako východziu mutáciu anglickú(en-US).

Pokiaľ vytvoríte novú jazykovú mutáciu Px Frameworku, môžete jej triedu zaslať mailom na emailovú adresu develop@proxima-soft.sk. Vývojový team Px Frameworku, potom doplní túto jazykovú mutáciu do budúcej distribúcie Px Farmeworku.

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 zadať 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 popisom, 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:PxEEdit ID="edtName" runat="server" CssClass="textbox" AddTableRow="True" ></Prx:PxEEdit>
```

V komponente PxEEdit, 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 PxEEdit v CSS deklarácii bude vyzerat' 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 PxEEdit vyzerat' nasledovne:

Editovať adresu

ID	29
Meno Osoby	Kveta Horvátová
Právna Forma	Fyzická osoba ▼
Dátum založenia	17.04.2010 
	☐ Adresa neplatná
Ulica	Spojová 34

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 zadať 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 popisom, 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 vyzerat' nasledovne:

```
<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;
}
.textbox-caption {
    background: lime;
    color: blue;
    font-size: larger;
    font-weight: bold;
    border: 0;
}
.textbox-button-insert
{
    width:20px;
    height:17px;
}
.textbox-button-delete
{
    width:20px;
    height: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 vyzeráť nasledovne:

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

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

Editovať adresu

ID	29	
Meno Osoby Dano	Kveta Horvátová	
Právna Forma	Právnická osoba	
Dátum založenia	10.04.2010	
	☐ Adresa neplatná	
Ulica	Spojová 34	
Kraj	Okres	Obec
Trenčiansky	Trenčín	Veľká Hradná
PSC	947 81	

+

Spojenie

Telefón

02/123456

-

Poznámka

pevná linka

+

Fax

Fax

02/ 123 456 82

-

Poznámka

fax

Ok

Storno

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 jednoliaty 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 vecí, 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 sekvencií pre danú databázovú platformu

V ďalšom budú uvedené vzory ConnectionStringu, pre danú databázovú platformu, a tiež ako správne vytvárať sekvencie (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 databázou 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.ConnectionString = "User Id=adr;Password=aa;Data Source=x;";
```

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

Na to aby ste prostredníctvom *PxWebQuery* vkladali nové riadky, potrebujete vytvoriť sekvenciu, z ktorej si komponenta *PxWebQuery* načíta jedinečné ID pre svoj primárny kľúč. Názov sekvencie, 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 sekvenciu 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 reťazecConnectionString. Dole je uvedený jeho vzor:

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

Ďalej musíme pre každú tabuľku, do ktorej sa budú zadávať nové riadky vytvoriť sekvenciu, 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 sekvenciu, z ktorej si komponenta *PxWebQuery* načíta jedinečné ID pre svoj primárny kľúč. MS SQL Server sekvencie nepodporuje, preto bolo zvolené riešenie, ktoré tieto sekvencie nahrádza. Teda ak chcete pracovať s Px Frameworkom pod databázou MS SQL, musíte si vytvoriť sekvenciu nasledujúcim spôsobom. Túto sekvenciu 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 nezmenený.

Tabuľku pre generovanie sekvencie, vytvoríte zadaním nasledujúceho SQL príkazu:

```
--VYTVORENIE SEQUENCIÍ PRE TABUĽKU ADRESAR A PRAVNA FORMA
--VYTVORENIE TABUĽKY, KDE SA BUDU UKLADAT HODNOTY SEQUENCIÍ
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_Sequence
    @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 @curvalue = @start

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

    set @result = @curvalue
    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.ConnectionString = "Database=dbname;Data Source=localhost;User Id=root;Password=aaaaa;"
```

Ďalej musíme pre každú tabuľku, do ktorej sa budú zadávať nové riadky vytvoriť sekvenciu, 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 sekvenciu, z ktorej si komponenta *PxWebQuery* načíta jedinečné ID pre svoj primárny kľúč. Mysql Server sekvencie nepodporuje, preto bolo zvolené riešenie, ktoré tieto sekvencie nahrádza. Teda ak chcete pracovať s Px Frameworkom pod databázou Mysql, musíte si vytvoriť sekvenciu nasledujúcim spôsobom. Túto sekvenciu 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 nezmenený.

Tabuľku pre generovanie sekvencie, vytvoríte zadáním 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 zadáním 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 delimitero.

```
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.ConnectionString = "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 nové 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.ConnectionString = "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 znakovkej 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 nové 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 prípade, 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 v.10.0.2531.0 SP1

MS SQL server 2005

MySQL server v.5.5.25. Win32

MySQL server v.5.1.46. Win32

MySQL server v.5.0.51b

FireBird server v.2.5.1.

FireBird server v.2.0.5.

InterBase server WI-V6.5.0.28.

5.1. 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.ConnectionString = "User Id=database01;Password=aa;Data Source=xe;";

    quPravForm.SQLText = "SELECT * FROM pravform";
    quPravForm.TimeOut = 5000; // sec
    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, musíme si do klauzuly uses zadať definovať assembly s názvom „Component“.

```
using Component;
```

V prípade dlhej odozvy servera je potrebné nastaviť property TimeOut. Doba čakania na odpoveď SQL servera sa zadáva v sekundách, defaultne je nastavená na 30 sekúnd.

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.