**Remove multiple items from the ListBox as in the following:**

**Step 1:** For this, we will first use some controls like TextBox (txtValue) to add the value in the ListBox, ListBox (lstValue) and an Add and Delete Button as in the following:

<input name="txtValue" type="text" />

<input type="button" name="add" value="Add" onclick="addValue();" />

<select name="lstValue" multiple>

<option value="empty"></select>

<input type="button" name="delete" value="Delete" onclick="deleteValue();" />

**Step 2:** Now we will write the code for adding the items in the ListBox like this:

<script language="javascript" type="text/javascript">

var i = 0;

function addValue() {

var v = document.form1.txtValue.value;

// get the TextBox Value and assign it into the variable

AddOpt = new Option(v, v);

document.form1.lstValue.options[i++] = AddOpt;

return true;

}

</script>

In this code:

document.form1.lstValue.options[i++] = AddOpt;

form1 is the name of the form and lstValue is the name of the ListBox and we will add the TextBox value in the form of the Option value (AddOpt) in the ListBox.

**Step 3:** Now we will write the code for deleting the value from the ListBox as in the following:

function deleteValue() {

var s = 1;

var Index;

if (document.form1.lstValue.selectedIndex == -1) {

alert("Please select any item from the ListBox");

return true;

}

while (s > 0) {

Index = document.form1.lstValue.selectedIndex;

if (Index >= 0) {

document.form1.lstValue.options[Index] = null;

--i;

}

else

s = 0;

}

return true;

}

In this Code

if (document.form1.lstValue.selectedIndex == -1) {

alert("Please select any item from the ListBox");

return true;

}

The following code will be used to determine whether there are any items in the LisBox:

while (s > 0) {

Index = document.form1.lstValue.selectedIndex;

if (Index >= 0) {

document.form1.lstValue.options[Index] = null;

--i;

}

else

s = 0;

}

Here we assign the SelectedIndex Value of the ListBox in the variable Index like this:

Index = document.form1.lstValue.selectedIndex;

And set the value Null in that index like this:

Here we select the Value Nidhi and delete it:

Now we will delete the Multiple Items from the ListBox like this:

So when we click on the Delete Button then the result will be:

Download 100% FREE Spire Office APIs

Add ListBox Items in Javascript

ListBox multiple Items remove

remove ListBox Items in Javascript Mahak Gupta

My name is Mahak Gupta. I am a C# Corner MVP.

http://www.c-sharpcorner.com/members/mahak-gupta

64

3.4m

2

×

Don’t miss Mahak Gupta’s next article Mahak Gupta

64

17k

3.4m

+ Follow

0

2

good one..

Mar 13, 2015

Gowtham Rajamanickam

42

21.6k

1.9m

0

0

Reply

great and simple...

Mar 13, 2015

Gowtham Rajamanickam

42

21.6k

1.9m

0

0

Reply

Comment Using File APIs for .NET

Aspose are the market leader of .NET APIs for file business formats – natively work with DOCX, XLSX, PPT, PDF, MSG, MPP, images formats and many more!

TRENDING UP 01

SharePoint 2013: Implement Deferred And Promise Object In JavaScript Asynchronous Execute

02

What's New In Visual Studio 2017

03

Learn MongoDB Step By Step

04

Experience The New Features In Visual Studio 2017

05

Visual Studio 2017 New Features

06

Basic Information And Overview Of Visual Studio 2017

07

Basics And Overview Of ASP.NET MVC Core or MVC 6 In Visual Studio 2017

08

Learn Basics Of MVC Using AngularJS

09

Introduction To Azure Machine Learning Studio

10

Basic Operations In SharePoint 2013 Using PowerShell

View All

Follow @twitterapi

Guess! What is coming here?

We will unveil at C# Corner Annual Conference 2017

About Us

Contact Us

Privacy Policy

Terms

Media Kit

Sitemap

Report a Bug

FAQ

©2017 C# Corner. All contents are copyright of their authors

Add (Insert) Items to ASP.Net ListBox using JavaScript

The following HTML Markup consists of an ASP.Net ListBox control, a TextBox and a Button.

When the Button is clicked, the AddValues JavaScript function is executed. Inside this function, first the value from the TextBox is fetched and then a HTML OPTION element is created.

The TextBox value is set to the InnerHtml and Value property of the OPTION element. Finally the OPTION element is appended to the ASP.Net ListBox control.

<asp:ListBox ID="ListBox1" runat="server" Width="150" Height="60"></asp:ListBox>

<br />

<hr />

<asp:TextBox ID="txtValue" runat="server" />

<asp:Button ID="btnAdd" Text="Add" runat="server" OnClientClick="return AddValues()" />

<script type="text/javascript">

function AddValues() {

var txtValue = document.getElementById("<%=txtValue.ClientID %>");

var listBox = document.getElementById("<%= ListBox1.ClientID%>");

var option = document.createElement("OPTION");

option.innerHTML = txtValue.value;

option.value = txtValue.value;

listBox.appendChild(option);

txtValue.value = "";

return false;

}

</script>

Add (Insert) Items to ASP.Net ListBox using jQuery

The following HTML Markup consists of an ASP.Net ListBox control, a TextBox and a Button.

When the Button is clicked, the jQuery click event handler is executed. Inside this event handler, first the value from the TextBox is fetched and then a HTML OPTION element is created.

The TextBox value is set to the InnerHtml and Value property of the OPTION element. Finally the OPTION element is appended to the ASP.Net ListBox control.

<asp:ListBox ID="ListBox1" runat="server" Width="150" Height="60" SelectionMode = "Multiple"></asp:ListBox>

<br />

<hr />

<asp:TextBox ID="txtValue" runat="server" />

<asp:Button ID="btnAdd" Text="Add" runat="server" />

<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>

<script type="text/javascript">

$(function () {

$("[id\*=btnAdd]").bind("click", function () {

var value = $("[id\*=txtValue]").val();

var listBox = $("[id\*=ListBox1]");

var option = $("<option />").val(value).html(value);

listBox.append(option);

$("[id\*=txtValue]").val("");

return false;

});

});

</script>

Screenshot

Fetching the ASP.Net ListBox items on Server Side (Code Behind)

The items added to the ASP.Net ListBox using JavaScript and jQuery are not added in ViewState and hence will not be available in the ListBox Items collection.

Thus we will need to make use of Request.Form collection and fetch the values in the following way.

C#

protected void Submit(object sender, EventArgs e)

{

string values = Request.Form[ListBox1.UniqueID];

}

VB.Net

Protected Sub Submit(sender As Object, e As System.EventArgs)

Dim values As String = Request.Form(ListBox1.UniqueID)

End Sub

The following screenshot displays the ListBox selected values posted to the server.

Introduction

JavaScript ListBox Control is a cross-browser JavaScript class. It is a sub set of HTML select element with attribute size >= 2.

Background

As you know, in the HTML ListBox element (select element with attribute size >=2), if you want to select multiple items, then you have to press & hold the Ctrl key during the process of selecting items. In a project, a client wanted a CheckBox against each item of the ListBox so that there was no need to press & hold Ctrl key while selecting multiple items. As HTML ListBox does not support CheckBox against each item, I've decided to develop my own custom ListBox control through JavaScript.

Constructor

ListBox Control has the following constructor:

ListBox(Arguments): The constructor of the ListBox class takes an argument of the type Object Literal. The definition of the argument Object Literal is given below:

Hide Copy Code

var Arguments = {

Base: \_Base, //Base reference where ListBox to be displayed.

Rows: \_Rows, //No. of visible items.

Width: \_Width, // Width of the ListBox.

NormalItemColor: \_NormalItemColor, // Normal item color.

NormalItemBackColor: \_NormalItemBackColor, // Normal item back color.

AlternateItemColor: \_AlternateItemColor, // Alternate item color.

AlternateItemBackColor: \_AlternateItemBackColor, // Alternate item back color.

SelectedItemColor: \_SelectedItemColor, // Selected item color.

SelectedIItemBackColor: \_SelectedIItemBackColor, // SelectedI item back color.

HoverItemColor: \_HoverItemColor, // Hover item color.

HoverItemBackColor: \_HoverItemBackColor, // Hover item back color.

HoverBorderdColor: \_HoverBorderdColor, // Hover bordered color.

ClickEventHandler: \_ClickEventHandler // Reference of the click event handler.

};

Example:

Hide Copy Code

var Arguments = {

Base: document.getElementById('base'),

Rows: 6,

Width: 300,

NormalItemColor: 'Black',

NormalItemBackColor: '#ffffff',

AlternateItemColor: 'Black',

AlternateItemBackColor: '#E0E0E0',

SelectedItemColor: '#ffffff',

SelectedIItemBackColor: '#E6A301',

HoverItemColor: '#ffffff',

HoverItemBackColor: '#2259D7',

HoverBorderdColor: 'orange',

ClickEventHandler: CheckBoxOnClick

};

You can assign each property of the argument Object Literal to null. In this case, each property will acquire its default value as:

Hide Copy Code

var Arguments = {

Base: null,

Rows: null,

Width: null,

NormalItemColor: null,

NormalItemBackColor: null,

AlternateItemColor: null,

AlternateItemBackColor: null,

SelectedItemColor: null,

SelectedIItemBackColor: null,

HoverItemColor: null,

HoverItemBackColor: null,

HoverBorderdColor: null,

ClickEventHandler: null

};

Properties & their default values have been tabulated below:Property Default Value

Base document.documentElement

Rows 6

Width 300

NormalItemColor 'Black'

NormalItemBackColor '#ffffff'

AlternateItemColor 'Black'

AlternateItemBackColor '#E0E0E0'

SelectedItemColor '#ffffff'

SelectedIItemBackColor '#E6A301'

HoverItemColor '#ffffff'

HoverItemBackColor '#2259D7'

HoverBorderdColor 'orange'

ClickEventHandler Anonymous method

Methods

ListBox Control has the following public methods:

AddItem(Text, Value): Used to add a ListBox Item. It takes two arguments:

Text: Item Text

Value:</strong /> Item Value

GetItems(): Used to get collection of all LBItem(ListBox Item)

GetItem(Index): Used to get a LBItem(ListBox Item) at a given Item index. Returns null in case Item isn't found. It takes one argument:

Index: Item Index.

DeleteItems(): Used to delete all the ListBox Items. Returns numbers of Items deleted

DeleteItem(Index): Used to delete a ListBox Item at a given Item index. Returns true on successful deletion, else false. It takes one argument:

Index: Item Index

GetTotalItems(): Used to get total number of ListBox Items

Contains(Index): Used to check whether a ListBox Item exists at a given Item index or not. Returns true if Item exists, else false. It takes one argument:

Index: Item Index

Dispose(): Used to destroy ListBox Object

Note: The LBItem is an Object Literal and has the following definition:

Hide Copy Code

var LBItem = {

IsSelected: \_IsSelected, // true/false.

Text: \_Text, // Item Text.

Value: \_Value, // Item Value.

ItemIndex: \_ItemIndex // Item Index.

};

Property

ListBox Control has only one property:

Version: Used to get the current version of the ListBox Control

Event

ListBox Control has only one event:

Click: Fires when any one Item CheckBox is clicked

The local anonymous method that responds to the click event (i.e. event handler) has the following signature:

Hide Copy Code

var EventHandlerName = function(Sender, EventArgs) {}

Here Sender is the reference of the element (in this case the Item CheckBox) that raises the click event & EventArgs is a Object Literal that contains necessary information regarding the event. EventArgs Object Literal has the following definition:

Hide Copy Code

var EventArgs = {

Text: \_Text, // Item Text.

Value: \_Value, // Item Value.

ItemIndex: \_ItemIndex // Item Index.

};

Using the Control

Add the reference of the ListBox.js file in your web page as:

Hide Copy Code

<script type="text/javascript" src="JS/ ListBox.js"></script>

Create a div element in the web page as:

Hide Copy Code

<div id="base"></div>

Now create a script tag in the head section of the web page & put the following code in the window.onload event as:

Hide Shrink Copy Code

<script type="text/javascript">

var oListBox;

window.onload = function()

{

var Arguments = {

Base: document.getElementById('base'),

Rows: 3,

Width: 300,

NormalItemColor: null,

NormalItemBackColor: null,

AlternateItemColor: null,

AlternateItemBackColor: null,

SelectedItemColor: null,

SelectedIItemBackColor: null,

HoverItemColor: null,

HoverItemBackColor: null,

HoverBorderdColor: null,

ClickEventHandler: OnClick

};

oListBox = new ListBox(Arguments);

oListBox.AddItem('CodeProject.com','http://www.codeproject.com');

oListBox.AddItem('yahoo.com','http://www.yahoo.com/');

oListBox.AddItem

('microsoft.com','http://www.microsoft.com/en/us/default.aspx');

oListBox.AddItem('asp.net','http://www.asp.net');

oListBox.AddItem('cricinfo.com','http://www.cricinfo.com/');

oListBox.AddItem('AOL','http://www.aol.com/');

oListBox.AddItem('STPL','http://stpl.biz');

}

</script>

In the above code, first an argument Object Literal with necessary properties has been created. After that ListBox has been instantiated using new keyword. Finally different ListBox Items have been added to the ListBox Object. Don't forget the click event wire up in the argument Object Literal as:

Hide Copy Code

ClickEventListener: OnClick

Where OnClick is the reference of the click event handler which is created as a local anonymous method:

Hide Copy Code

var OnClick = function(Sender, EventArgs)

{

//Code

}

Example:

Hide Copy Code

var OnClick = function(Sender, EventArgs)

{

var Message = new Array();

Message.push('IsSelected: ' + Sender.checked.toString());

Message.push('Text: ' + EventArgs.Text);

Message.push('Value: ' + EventArgs.Value);

Message.push('Index: ' + EventArgs.ItemIndex);

document.getElementById('DivMessage').innerHTML = Message.join('<br />');

}

This method will get called when you will click on any one Item CheckBox of the ListBox control.

Invoke the Dispose method in the window.onunload event in order to destroy ListBox Object as:

Hide Copy Code

window.onunload = function(){oListBox.Dispose(); }

Conclusion

So this is my approach to develop custom JavaScript ListBox control. Although it is only a subset of existing HTML ListBox element, it is more user friendly than the existing one. It can be further customized for different requirements. Please let me know about bugs and/or errors & give suggestions to improve this ListBox control.

Browser Compatibility

I have tested this control on a number of web browsers.

ListBoxItems Class

Members Collapse All

Language Filter: All Syncfusion.JavaScript.Web Namespace : ListBoxItems Class

>

Syntax

Visual Basic

C#

JScript

C++/CLI

'Declaration

<ParseChildrenAttribute(ChildrenAsProperties=True, DefaultProperty="Items")>

<ToolboxItemAttribute(ToolboxItemType=, ToolboxItemTypeName="")>

<SerializableAttribute()>

Public Class ListBoxItems

'Usage

Dim instance As ListBoxItems

Inheritance Hierarchy

System.Object

Syncfusion.JavaScript.Web.ListBoxItems

AssemblyVersion

Syncfusion.EJ.Web: 15.1451.0.37

See Also

Reference

ListBoxItems Members

Syncfusion.JavaScript.Web Namespace

Send feedback regarding this topic

Listbox options javascript select all,move left-right, move up-down

by Viral Patel · June 15, 2009

While working with Listboxes I had to write few small JavaScript snippets to perform certain tasks like selecting all options / seselecting all options or moving options up and down or swapping the options between two listboxes. I thought of sharing the JavaScript functions with you so that you can bookmark the article and use the code whenever you need them.

Listbox Select All / Deselect All JavaScript

Following is the JavaScript function for implementing select all, deselect all options in listbox.

function listbox\_selectall(listID, isSelect) {

var listbox = document.getElementById(listID);

for(var count=0; count < listbox.options.length; count++) {

listbox.options[count].selected = isSelect;

}

}

The above javascript code is very straight forward. All you have to do it to pass the ID of listbox you need to perform select all/deselect all and a boolean value for select/deselect.

For example if a listbox has an id “countryList” following can be the function call.

listbox\_selectall('countryList', true); //select all the options

listbox\_selectall('countryList', false); //deselect all the options

Listbox Move up/down options JavaScript

Following is the javascript function that you can use to add move up/down options functionality. User can select any option and move it up in the list or down.

function listbox\_move(listID, direction) {

var listbox = document.getElementById(listID);

var selIndex = listbox.selectedIndex;

if(-1 == selIndex) {

alert("Please select an option to move.");

return;

}

var increment = -1;

if(direction == 'up')

increment = -1;

else

increment = 1;

if((selIndex + increment) < 0 ||

(selIndex + increment) > (listbox.options.length-1)) {

return;

}

var selValue = listbox.options[selIndex].value;

var selText = listbox.options[selIndex].text;

listbox.options[selIndex].value = listbox.options[selIndex + increment].value

listbox.options[selIndex].text = listbox.options[selIndex + increment].text

listbox.options[selIndex + increment].value = selValue;

listbox.options[selIndex + increment].text = selText;

listbox.selectedIndex = selIndex + increment;

}

Thus, you can call this function with first argument as list id and second argument a string value ‘up’ or ‘down’ depending on where you want to move the selected option.

listbox\_move('countryList', 'up'); //move up the selected option

listbox\_move('countryList', 'down'); //move down the selected option

Listbox swap/move left-right options JavaScript

Following is the javascript code for moving selected options from one listbox to another.

function listbox\_moveacross(sourceID, destID) {

var src = document.getElementById(sourceID);

var dest = document.getElementById(destID);

for(var count=0; count < src.options.length; count++) {

if(src.options[count].selected == true) {

var option = src.options[count];

var newOption = document.createElement("option");

newOption.value = option.value;

newOption.text = option.text;

newOption.selected = true;

try {

dest.add(newOption, null); //Standard

src.remove(count, null);

}catch(error) {

dest.add(newOption); // IE only

src.remove(count);

}

count--;

}

}

}

Just call the function with two arguments, first is the source listbox id and second is the destination listbox id.

listbox\_moveacross('countryList', 'selectedCountryList');

Online Demo

Click here for the online demo.

Getting Values in Server side script

When you submit the form with Listbox, by default browser will submit only selected values from that selectbox. Thus if you want to submit values from right hand side select box, you first needs to select all those before form gets submitted.

You can do this like below:

<script>

function submit() {

listbox\_selectall('righthand\_side\_listbox\_id', true);

return true;

}

<form onsubmit="return submit()">

...

</form>

This would select all the options before form gets submitted. Now you can retrieve options in your server side script from request as request parameter.

Related Articles

Deleting Multiple Values From Listbox in JavaScript

21 JavaScript Tips and Tricks for JavaScript Developers

Multiple Selection Listbox Javascript Problem in MS Internet Explorer

Dynamic combobox-listbox-drop-down using javascript

Multiple Checkbox Select/Deselect using jQuery – Tutorial with Example

Dynamically add button, textbox, input, radio elements in html form using JavaScript.

Tutorial: Handle browser events using jQuery JavaScript framework

Add ListBoxItem to ListBox from C# code behind using a javascript function Rate this:

See more: C#

ASP.NET

Javascript

HTML

ListBox

I have a page that's written using ASP.Net (C#)

I have a button that is NOT a server control but is just an anchor (

Hide Copy Code

<a href="java<!-- no -->script:Add\_Button\_Click">Add</a>

).

The menu code looks like this:

Hide Copy Code

<ul>

<li><a href="java<!-- no -->script:Add\_Button\_Click('someString');">Add Value To ASP.Net ListBox</a></li>

</ul>

The "Add\_Button\_Click" (JavaScript):

Hide Copy Code

<script type="text/javascript">

function Add\_Button\_Click(someStringValue) {

}

</script>

The "Add\_Button\_Click" (C# Code Behind):

Hide Copy Code

public static void Add\_Button\_Click(string someValue)

{

ListBox box = Page.FindControl("ListBox1") as ListBox;//I'm not too sure if this part works but it's what I have so far.

}

In my page's Code Behind I need to be able to do two things:

1.) Call a C# function from the JavaScript code.

2.) In the C# function that's being called from JavaScript I need to add a ListBoxItem to an ASP.Net ListBox Server Control that resides on my page.

I've searched the web but can't seem to find a way to make any of these things work correctly.

Please show/tell me with example code how you guys would go about doing this, Thanks!!!

Add items in Drop Down List or List Box using Javascript

June 6, 2007 by chiragrdarji

I have seen lots of questions in diffrent forums for adding items in drop down list or list box using javascript. Below is the script for the same.

<script type="text/javascript">

function AddItem(Text,Value)

{

// Create an Option object

var opt = document.createElement("option");

// Add an Option object to Drop Down/List Box

document.getElementById("DropDownList").options.add(opt);

// Assign text and value to Option object

opt.text = Text;

opt.value = Value;

}

<script />

You can use this function in for loop to add more than one item.

Happy Programming !!

Advertisements

Rate this:

101 Votes

Related

Moving Options/values up and down in list box using javascript

In "DotNet 2.0"

Debugging JavaScript in MAC Safari / Tool to debug JavaScript in Safari

In "DotNet 2.0"

Thousand Separator function for Java Script

In "DotNet 3.0"

Posted in DotNet 2.0, Javascript, Techincal | 176 Comments

176 Responses

on June 14, 2007 at 12:02 pm | Reply

anil

yah its really nice solution…

that will help me in ajax application

on December 21, 2009 at 4:58 pm | Reply

Zulfi

Hi,

Please let me know about this issue:

I have a dropdown with the following options in it.

Select

New Category

abc

efg

and when the user selects new category he will be given by a text field to enter the calue for the new category; and the value entered by the user should be added in the dropdown list as the last time.

for eg: if user select new actegory and enters the value as “ghj” then this value should be the last item in the dropdown and it should lok something like:

Select

New Category

abc

efg

ghj

Please help me out with this.

Thanks,

Zulfi

Add, Delete Items in DropDownList, ListBox using Javascript

I have seen lots of questions in Asp.Net forums for adding/deleting items in drop down list or list box using JavaScript. Here is the code...

<asp:DropDownList ID="DropDownList1" runat="server" Width="182px">

<asp:ListItem value="1" Text ="Approve"></asp:ListItem>

<asp:ListItem value="2" Text ="Accept"></asp:ListItem>

<asp:ListItem value="3" Text ="Test1"></asp:ListItem>

<asp:ListItem value="4" Text ="Test2"></asp:ListItem>

</asp:DropDownList>

<input type="button" value="Remove selected item" onclick="JavaScript: DeleteItem();" />

<input type="text" id="ddlText" name="ddlText" />

<input type="text" id="ddlValue" name="ddlValue" />

<input type="button" value="Add item" onclick="JavaScript: AddItem();" />

<input type="hidden" id="ddlElements" name="ddlElements" runat="server" />

<asp:Button ID="Button1" runat="server" Text="Button" />

<script type="text/javascript">

function DeleteItem()

{

var dropDownListRef = document.getElementById('<%= DropDownList1.ClientID %>');

var optionsList = '';

if ( dropDownListRef.value.length > 0 )

{

var itemIndex = dropDownListRef.selectedIndex;

if ( itemIndex >= 0 )

dropDownListRef.remove(itemIndex);

}

else

{

alert('Please select an item');

dropDownListRef.focus();

dropDownListRef.select();

}

for (var i=0; i<dropDownListRef.options.length; i++)

{

var optionText = dropDownListRef.options[i].text;

var optionValue = dropDownListRef.options[i].value;

if ( optionsList.length > 0 )

optionsList += ';';

optionsList += optionText;

optionsList += ';';

optionsList += optionValue;

}

document.getElementById('<%= ddlElements.ClientID %>').value = optionsList;

}

function AddItem()

{

var dropDownListRef = document.getElementById('<%= DropDownList1.ClientID %>');

var ddlTextRef = document.getElementById('ddlText');

var ddlValueRef = document.getElementById('ddlValue');

var optionsList = '';

if ( ddlTextRef.value !="" && ddlValueRef.value!="" )

{

var option1 = document.createElement("option");

option1.text= ddlValueRef.value;

option1.value= ddlTextRef.value ;

dropDownListRef.options.add(option1);

}

else

alert('Please enter values');

for (var i=0; i<dropDownListRef.options.length; i++)

{

var optionText = dropDownListRef.options[i].text;

var optionValue = dropDownListRef.options[i].value;

if ( optionsList.length > 0 )

optionsList += ';';

optionsList += optionText;

optionsList += ';';

optionsList += optionValue;

}

document.getElementById('<%= ddlElements.ClientID %>').value = optionsList;

}

</script>

In Code behind Page\_Load, add following code..

if (IsPostBack)

{

DropDownList1.Items.Clear();

string[] DropDownListArray = ddlElements.Value.Trim().Split(';');

for (int i = 0; i < DropDownListArray.Length; i = i + 2)

{

string itemText = DropDownListArray[i];

string itemValue = DropDownListArray[i + 1];

DropDownList1.Items.Add(new ListItem(itemText, itemValue));

}

}

string optionsList = string.Empty;

for (int i = 0; i < DropDownList1.Items.Count; i++)

{

string optionText = DropDownList1.Items[i].Text;

string optionValue = DropDownList1.Items[i].Value;

if (optionsList.Length > 0)

optionsList += ";";

optionsList += optionText;

optionsList += ';';

optionsList += optionValue;

}

ddlElements.Value = optionsList;

Update: Client-side changes to a DropDownList/ListBox are not persisted server-side, so any changes made will be lost if a PostBack occurs. Added server-side persistence to code. (Thanks to tip form NC01)

Tags:

Categories: ASP.NET | JavaScript | Web Development

Posted by vijay on 14 декабря 2007 г. 21:40

E-mail | Kick it! | DZone it! | del.icio.us

Permalink | Комментарии (0) | Post RSS

Похожие записи

Limit number of items to be selected in Asp.Net ListBox

Here is an example to validate ListBox. This validation will check user to select less than10 items&...

Asp.Net Listbox–Get Selected items to List<string>

List<string> liItems= (from ListItem item in ListBox1.Items where item.Selected s...

Click event for Listbox

In Asp.Net ListBox, there are no events for Click action. The follwing code is useful in this rega...

How do you add item in listbox through the textbox?

Advertisement

Edit

Answered by The WikiAnswers® Community

Answers.com® is making the world better one answer at a time.

This can be done with a little javascript. Here's an example <html>

<head>

<script type="text/javascript" language="javascript">

function addNewItem()

{

// Retrieve the elements from the document body

var textbox = document.getElementById('MyTextbox');

var listbox = document.getElementById('MyListbox'); // Now we need to create a new 'option' tag to add to MyListbox

var newOption = document.createElement('option');

newOption.value = textbox.value; // The value that this option will have

newOption.innerHTML = textbox.value; // The displayed text inside of the <option> tags // Finally, add the new option to the listbox

listbox.appendChild(newOption);

}

</script>

</head> <body> <input id="MyTextbox" type="textbox" />

<input type="button" value="Add Item" onclick="javascript:addNewItem()" /> <br /><br /> <select id="MyListbox" size="10">

<option value="apples">Apples</option>

<option value="oranges">Oranges</option>

<option value="bananas">Bananas</option>

</select> </body>

</html>