

Agenda

1. *What is AJAX*
2. *What is AJAX.NET*
3. *Script Manager*
4. *Update Panel*
5. *Update Progress*
6. *Timer*
7. *AJAX Control Toolkit*

Deccansoft

AJAX

Default.aspx

```
<script language="javascript" type="text/javascript">
    var objXmlHttp
    function btnGetServerTime_onclick() {
        objXmlHttp = new ActiveXObject("Msxml2.XMLHTTP")
        objXmlHttp.onreadystatechange = StateChanged
        objXmlHttp.open("GET", "RenderServerTime.aspx", true) //To Establish connection with the webserver.
        objXmlHttp.send(null) //Submits the request to the webserver.
    }
    //It's a callback function executed when the response of the page is received by the browser.
    function StateChanged() {
        if (objXmlHttp.readyState == 4 || objXmlHttp.readyState == "complete")
            document.forms[0].TextBox1.value = objXmlHttp.responseText;
    }
</script>
</head>
<body>
    <form id="form1" runat="server">
        <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
        <input type="button" value="Get Server Time" onclick="btnGetServerTime_onclick()" /></form>
</body>
</html>
```

In Page_Load of RenderServerTime.aspx

```
protected void Page_Load(object sender, EventArgs e)
{
    Response.Clear();
    Response.Cache.SetCacheability(HttpCacheability.NoCache);
    Response.Write(DateTime.Now.ToString());
    Response.End();
}
```

AJAX.NET

Ajax.Net is AJAX framework integrated with ASP.NET framework.

It has following controls:

1. ScriptManager – Must be preset on every page which is using other AJAX controls.
2. UpdatePanel – Container for fragment of the page which should be asynchronously posted when required.
3. UpdateProgress – Container for showing something when asynchronous postback is in progress.
4. Timer – Refreshing a fragment of the page at a regular interval of time.

Update Panel

UpdatePanel.aspx

```
<div>
  <asp:DropDownList ID="ddlDemo2" runat="server" AutoPostBack="True"
OnSelectedIndexChanged="ddlDemo_SelectedIndexChanged">
    <asp:ListItem>One-2</asp:ListItem>
    <asp:ListItem>Two-2</asp:ListItem>
    <asp:ListItem>Three-2</asp:ListItem>
    <asp:ListItem>Four-2</asp:ListItem>
    <asp:ListItem>Five-2</asp:ListItem>
    <asp:ListItem>Six-2</asp:ListItem>
  </asp:DropDownList>
  <asp:ScriptManager ID="ScriptManager1" runat="server">
  </asp:ScriptManager>
  <asp:UpdatePanel runat="server" ID="up1">
    <ContentTemplate>
      <asp:DropDownList ID="ddlDemo" runat="server" AutoPostBack="True"
OnSelectedIndexChanged="ddlDemo_SelectedIndexChanged">
        <asp:ListItem>One</asp:ListItem>
        <asp:ListItem>Two</asp:ListItem>
        <asp:ListItem>Three</asp:ListItem>
        <asp:ListItem>Four</asp:ListItem>
        <asp:ListItem>Five</asp:ListItem>
        <asp:ListItem>Six</asp:ListItem>
      </asp:DropDownList>
      <asp:UpdateProgress ID="UpdateProgress1" runat="server" AssociatedUpdatePanelID="up1">
        <ProgressTemplate>
          Please Wait...
        </ProgressTemplate>
      </asp:UpdateProgress>
    </ContentTemplate>
  </asp:UpdatePanel>
</div>
```

```

<br />
<br />
<asp:Label ID="lblDemo" runat="server" Text=""></asp:Label>
</ContentTemplate>
<Triggers>
  <asp:AsyncPostBackTrigger ControlID="ddlDemo2" EventName="SelectedIndexChanged" />
</Triggers>
</asp:UpdatePanel>
<br />
Some other content...
<br />
<asp:UpdatePanel ID="up2" runat="server">
  <ContentTemplate>
    <asp:Label ID="lblDemo2" runat="server" Text=""></asp:Label>
    <br />
    <br />
    <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Show Alert" />
  </ContentTemplate>
</asp:UpdatePanel>
<br />
<b>Timer Demo:<br /></b>
<asp:UpdatePanel runat="server" ID="upTimer">
  <ContentTemplate>
    <asp:Label ID="lblServerTime" runat="server"></asp:Label>
    <asp:Timer ID="Timer1" runat="server" Interval="5000" OnTick="Timer1_Tick">
    </asp:Timer>
  </ContentTemplate>
</asp:UpdatePanel>
</div>

```

UpdatePanel.aspx.cs

```

protected void Page_Load(object sender, EventArgs e)
{
    lblServerTime.Text = DateTime.Now.ToLongTimeString();
}

protected void ddlDemo_SelectedIndexChanged(object sender, EventArgs e)
{
    System.Threading.Thread.Sleep(5000);
    lblDemo.Text = ddlDemo.SelectedValue;
    lblDemo2.Text = ddlDemo.SelectedValue;
}

```

```

}

protected void Button1_Click(object sender, EventArgs e)
{
    //Page.ClientScript.RegisterStartupScript(this.GetType(), "k1", "alert('Hello');", true);
    ScriptManager.RegisterStartupScript(up2, up2.GetType(), "k1", "alert('Hello');", true);
}

protected void Timer1_Tick(object sender, EventArgs e)
{
    lblServerTime.Text = DateTime.Now.ToLongTimeString();
}

```

Note: If the button (in update panel) is clicked and if the click event handler on server has to render some javascript, it should as below:

```
ScriptManager.RegisterStartupScript(upId, upId.GetType(), "key1", "alert('Hello');", true);
```

Script Manager

```

<asp:ScriptManager runat="server" ID="MainScriptManager" EnableHistory="true"
    OnNavigate="MainScriptManager_Navigate" EnableSecureHistoryState="false" />

protected void ddlColor_SelectedIndexChanged(object sender, EventArgs e)
{
    lblSelectedColor.Text = ddlColor.SelectedValue;

    lblSelectedColor.BackColor = System.Drawing.Color.FromName(ddlColor.SelectedValue);

    MainScriptManager.AddHistoryPoint("SelectedColor", ddlColor.SelectedValue);
}

protected void MainScriptManager_Navigate(object sender, HistoryEventArgs e)
{
    string color = e.State["SelectedColor"];

    if (StringUtil.IsNullOrEmpty(color))
    {
        lblSelectedColor.Text = color;

        lblSelectedColor.BackColor = System.Drawing.Color.FromName(color);
    }
}

```

AJAX Control Toolkit

1. Visit <http://ajaxcontroltoolkit.codeplex.com/>
2. Download the file - AjaxControlToolkit.Binary.NET4.zip and extract in your local drive.
3. File → Open → Website → File System → d:\...\SampleWebSite
4. Run the application and walkthrough all the samples...

Steps to use AJAXControlToolKit in the Web Application Project.

1. ToolBox → Right Click → Add Tab (AjaxControlToolKit)
2. ToolBox → Right Click → Choose Items
3. Browse →\SampleWebSite\bin\AjaxControlToolKit.Dll → OK

Demos of how to use Ajax Toolkit controls.

AutoComplete

1. Add a new webform to the project
2. Drag and Drop ASP.NET ScriptManager on it.
3. Drag and Drop TextBox (txtDemo)
4. Drag and Drop AutoCompleteExtender
5. From SampleWebSite, AutoComplete sample copy the tag <AutoCompleteExtender....>
6. Paste the same in our webform. (replace tagprefix of AutoCompleteExtender)
7. Replace TargetControlID with Id of our TextBox (txtDemo).
8. Add to the Project a WebService (AutoComplete.asmx)
9. Uncomment / add the attribute "[System.Web.Script.Services.ScriptService()]" in the class of file "App_code\ AutoComplete.cs"
10. To the WebService add a WebMethod "GetCompletionList" as in example below.

AutoComplete.aspx

```
<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="cc1" %>
<form id="form1" runat="server">
  <div>
    <asp:ScriptManager ID="ScriptManager1" runat="server">
    </asp:ScriptManager>
    <asp:TextBox ID="txtDemo" runat="server"></asp:TextBox>
  </div>
  <cc1:AutoCompleteExtender ID="AutoCompleteExtender1" runat="server" TargetControlID="txtDemo"
ServicePath="AutoComplete.asmx" ServiceMethod="GetCompletionList" MinimumPrefixLength="1"
  CompletionInterval="1000" EnableCaching="true" CompletionSetCount="12">
  </cc1:AutoCompleteExtender>
</form>
```

AutoComplete.asmx

```
<%@ WebService Language="VB" CodeBehind="~/App_Code/AutoComplete.vb" Class="AutoComplete" %>
```

App_Code/AutoComplete.cs

```
[System.Web.Script.Services.ScriptService()]  
public class AutoComplete : System.Web.Services.WebService  
{  
    [WebMethod]  
    public string[] GetCompletionList(string prefixText, int count)  
    {  
        //"Select EmpName from Emp where EmpName like '" + prefixText + "%"  
        string[] str = new string[5];  
        str[0] = prefixText + "1";  
        str[1] = prefixText + "2";  
        str[2] = prefixText + "3";  
        str[3] = prefixText + "4";  
        str[4] = prefixText + "5";  
        return str;  
    }  
}
```

ModalPopup

ModalPopup.aspx

```
<asp:Button ID="btnModalPopup" runat="server" Text="Modal Popup" /><br />  
    <asp:Panel ID="Panel1" runat="server" Height="50px" Width="50%" Style="display: none">  
        <pre>This is Modal Popup Demo</pre>  
        <asp:Button ID="btnOk" runat="server" Text="Ok" Width="75px" />  
    </asp:Panel>  
<cc1:ModalPopupExtender ID="ModalPopupExtender1" runat="server" TargetControlID="btnModalPopup"  
    PopupControlID="Panel1" BackgroundCssClass="modalBackground" OkControlID="btnOk" Drag="true"/>
```

Accordion

Accordion.aspx

```
<cc1:Accordion ID="MyAccordion" runat="server" SelectedIndex="0" AutoSize="None">  
    <Panes>  
        <cc1:AccordionPane ID="AccordionPane1" runat="server">
```

```

<Header>1.MS.NET</a></Header>

<Content>
    1.VB.NET<br />
    2.C#<br />
    3.ASP.NET<br />
</Content>
</cc1:AccordionPane>
<cc1:AccordionPane ID="AccordionPane2" runat="server">
    <Header>2. Java</a></Header>
    <Content>
        1.Core<br />
        2.Advance<br />
        3.J2EE<br />
    </Content>
</cc1:AccordionPane>
</Panels>
</cc1:Accordion>

```

CascadingDropDown.aspx

CascadingDropDown.aspx

```

Country : <asp:DropDownList ID="ddlCountry" runat="server" Width="180px"> </asp:DropDownList>
State : <asp:DropDownList ID="ddlState" runat="server" Width="180px"></asp:DropDownList>
City : <asp:DropDownList ID="ddlCity" runat="server" Width="180px" AutoPostBack="True"
OnSelectedIndexChanged="ddlCity_SelectedIndexChanged" ></asp:DropDownList>

<cc1:CascadingDropDown ID="cddlCountry" runat="server" TargetControlID="ddlCountry" Category="Country"
PromptText="Please select a Country" LoadingText="[Loading Countries...]"
ServiceMethod="GetDropDownContents" />

<cc1: CascadingDropDown ID="cddlState" runat="server" TargetControlID="ddlState" Category="State"
PromptText="Please select a State" LoadingText="[Loading States...]" ServiceMethod="GetDropDownContents"
ParentControlID="ddlCountry" />

<cc1: CascadingDropDown ID="cddlCity" runat="server" TargetControlID="ddlCity" Category="City"
PromptText="Please select a City" LoadingText="[Loading Cities...]" ServiceMethod="GetDropDownContents"
ParentControlID="ddlState" />

```


CascadingDropDown.aspx.cs

```
using System;
using AjaxControlToolkit;
[System.Web.Services.WebMethod()]
public static CascadingDropDownNameValue[] GetDropDownContents(string knownCategoryValues, string
category)
{
    CascadingDropDownNameValue[] arCDD = new CascadingDropDownNameValue[3];
    StringDictionary dic = CascadingDropDown.ParseKnownCategoryValuesString(knownCategoryValues);
    if (category == "country")
    {
        arCDD[0] = new CascadingDropDownNameValue("C1", "1");
        arCDD[1] = new CascadingDropDownNameValue("C2", "2");
        arCDD[2] = new CascadingDropDownNameValue("C3", "3");
    }
    else if (category == "state")
    {
        arCDD[0] = new CascadingDropDownNameValue(dic["country"] + "-S1", "1");
        arCDD[1] = new CascadingDropDownNameValue(dic["country"] + "-S2", "2");
        arCDD[2] = new CascadingDropDownNameValue(dic["country"] + "-S3", "3");
    }
    else if (category == "city")
    {
        arCDD[0] = new CascadingDropDownNameValue(dic["country"] + "-" + dic["state"] + "-C1", "1");
        arCDD[1] = new CascadingDropDownNameValue(dic["country"] + "-" + dic["state"] + "-C2", "2");
        arCDD[2] = new CascadingDropDownNameValue(dic["country"] + "-" + dic["state"] + "-C3", "3");
    }
    return arCDD;
}
```

Note: In the above example the WebMethod is placed in the webform and not in the webservice it is also declared as static and has the attribute "System.Web.WebService.WebMethod()".