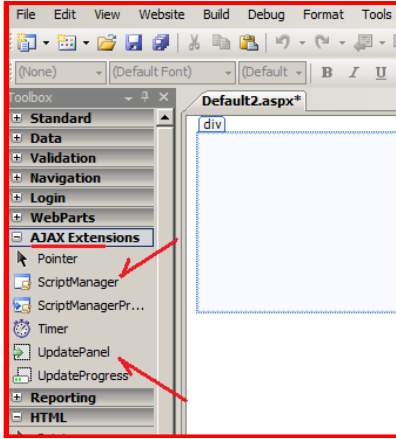


## AJAX KULLANIMI

Web sayfaları çalışırken Server a her gidişte yeni sayfanın tüm bilgileri yeniden yüklenir. Bu durum hem serverdan alınan bilgilerin fazla olmasını (dolayısı ile download trafiğini artırır) hemde sayfanın görüntülenme zamanını uzatır. Oysa çoğunlukla her sayfa yenilemede sayfa üzerindeki çok az bir kısmın bilgisinin değişmesi istenir. Bunu sağlamak için Ajax kullanılır.

Ajax'ı sayfamızda kullanabilmek için öncelikle bu işleri kontrol için "**ScriptManager**" nesnesinin sayfamızda olması gerekir. Bu nesneler Ajax komut grubu içinde bulunmaktadır.



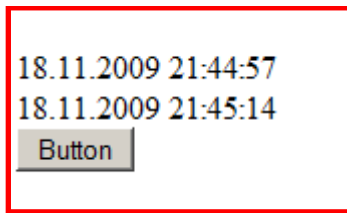
```
<asp:ScriptManager ID="ScriptManager1" runat="server">
</asp:ScriptManager>
```

Sayfanın tamamı yerine sadece ilgili nesne üzerindeki bilgilerin güncellemek için "**UpdatePanel**" etiketi kullanılır. Yani güncelleme işlemi bu panel içindeki nesnelere gerçekleştirilir. Sayfada bulunan bu panelin dışındaki nesnelere içindeki bilgiler yeniden okunmaz.

```
<asp:UpdatePanel ID="UpdatePanel1" runat="server">
    <ContentTemplate>
        <asp:Label ID="Label2" runat="server"></asp:Label>
        <br />
        <asp:Button ID="Button1" runat="server"
onclick="Button1_Click" Text="Button" />
    </ContentTemplate>
</asp:UpdatePanel>
```

Ajax kullanımı ile ilgili olarak aşağıdaki örnekleri inceleyelim.

## AJAX ÖRNEK 1:



### Default.aspx

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

```
<!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>
```

```
<asp:ScriptManager ID="ScriptManager1" runat="server">
```

```
</asp:ScriptManager>
```

```
<br />
```

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

```
<br />
```

```
<asp:UpdatePanel ID="UpdatePanel1" runat="server">
```

```
<ContentTemplate>
```

```
<asp:Label ID="Label2" runat="server"></asp:Label>
```

```
<br />
```

```
<asp:Button ID="Button1" runat="server"
```

```
onclick="Button1_Click" Text="Button" />
```

```
</ContentTemplate>
```

```
</asp:UpdatePanel>
```

```
<br />
```

```
</div>
```

```
</form>
```

```
</body>
```

```
</html>
```

### Default.aspx.cs

```
using System;  
using System.Configuration;  
using System.Data;  
using System.Linq;  
using System.Web;  
using System.Web.Security;  
using System.Web.UI;  
using System.Web.UI.HtmlControls;
```

```

using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

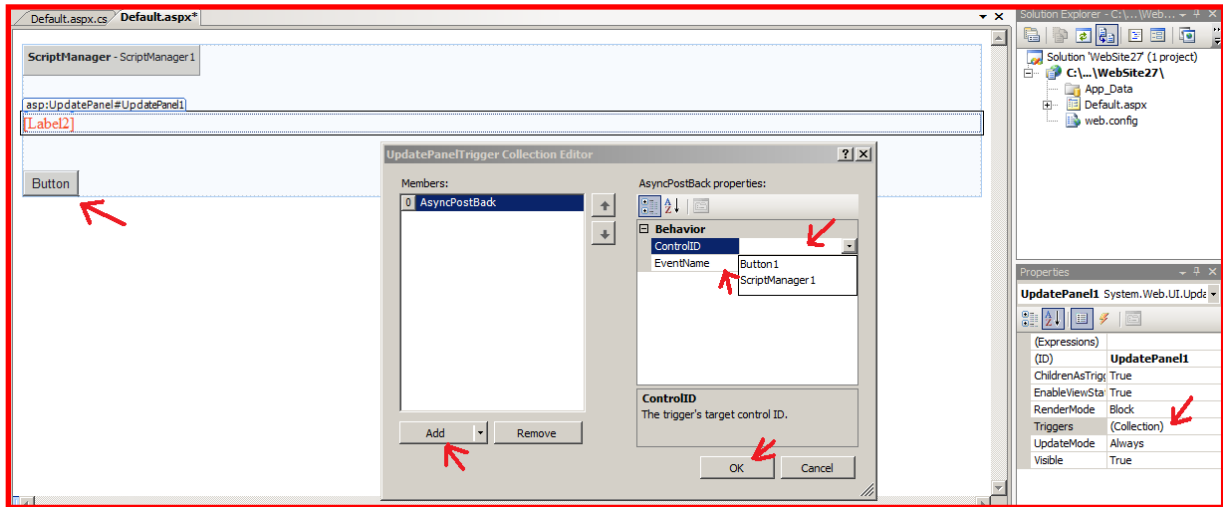
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        Label1.Text = DateTime.Now.ToString() ; //Burayı güncellemez çünkü
        label1 Update panelin içinde değildir.
        Label2.Text = DateTime.Now.ToString(); //buton ve label2 update
        panelin içinde olduğu için burasını her tıkladığımızda günceller. Dikkat
        sayfa güncellenmese bile Load olayının içindeyken burayı güncelliyebiliyor.
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
    }
}

```

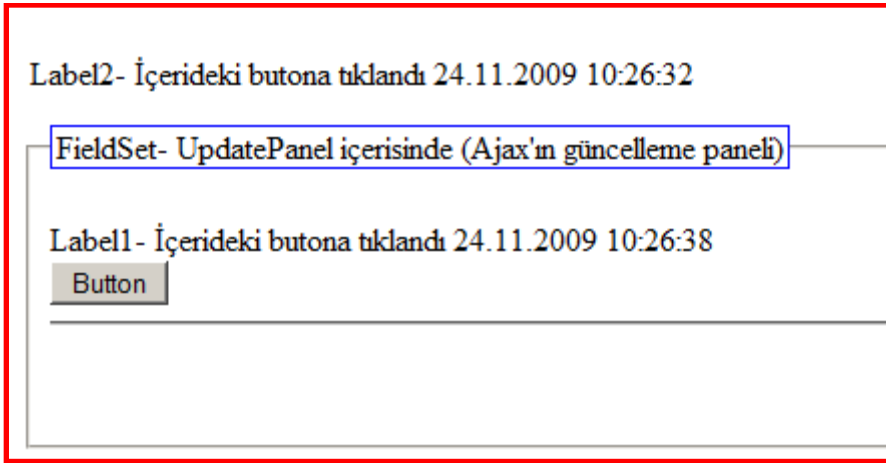
**Not:**

Eğer buton UpdatePanelin dışında olursa Yine güncelleme yaptırabiliriz. Bu durumda Update Panelin Triggers (tetikleyiciyi) ayarlamalıyız. Aşağıdaki şekilde yapabiliriz.



## AJAX ÖRNEK 2:

### Ajax kodu yazmadan basit bir şekilde sayfa içerisindeki bilgilerin güncellenmesi



Burada Buton ve Label 1 Update panelin içerisindedir. Label 2 ise dışındadır. Butona tıkladığımızda Serveraa sayfanın tamamı gitmiyor sadece Update panelin içeriği gidiyor ve dolayısıyla sadece orası güncelleniyor. Sebebi olaya sebep olan butona update panelin içerisinde olduğu için sayfanın tamamını göndermiyor. (Acaba buton update panelin dışında olsa sayfanın tamamı gidermi ?)

### Default.aspx

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

<!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">
    <p>
      <br />
      &nbsp;<asp:Label ID="Label2" runat="server"
Text="Label"></asp:Label></p>
    <p>
      <asp:ScriptManager ID="ScriptManager1" runat="server">
        </asp:ScriptManager>
    </p>

    <fieldset>
      <legend style="border: 1px solid #0000FF">FieldSet- UpdatePanel
içerisinde (Ajax'ın güncelleme paneli)</legend>

      <br />

      <asp:UpdatePanel ID="UpdatePanel1" runat="server">
        <ContentTemplate>

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

        <br />
    </asp:UpdatePanel>
  </form>
</body>
</html>
```

```

        <asp:Button ID="Button3" runat="server" Text="Button" />
        <hr />
        <br />

        </ContentTemplate>
        </asp:UpdatePanel>
        <br />

    </fieldset>

</form>
</body>
</html>

```

## Default.aspx.cs

```

using System;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        Label1.Text = "Label1- İçerideki butona tıklandı " +
DateTime.Now.ToString();
        Label2.Text = "Label2- İçerideki butona tıklandı " +
DateTime.Now.ToString();
    }
}

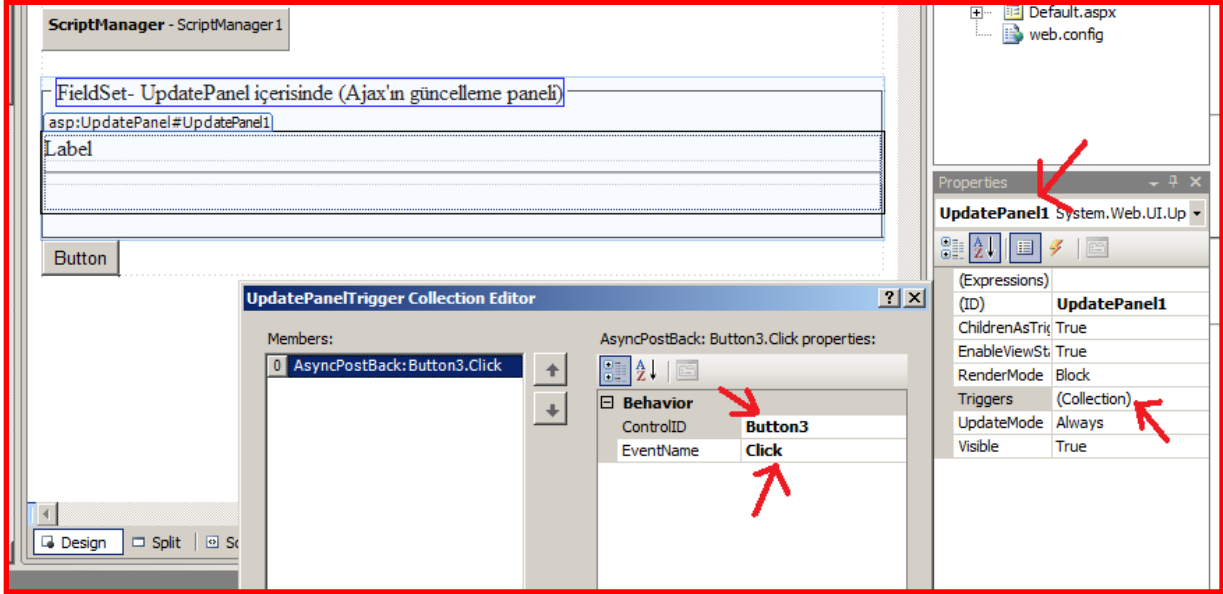
```

**NOT:** Bu örnekte butonu Update panelin dışına aldığımızda sayfanın tamamı servera gidecektir. Bu sebeple load kısmındaki kodlar sebebiyle her iki label ıda güncelleyecektir. Sayfanın servera gidip gitmediğini Browserın altındaki zaman göstergesinden anlayabiliriz.



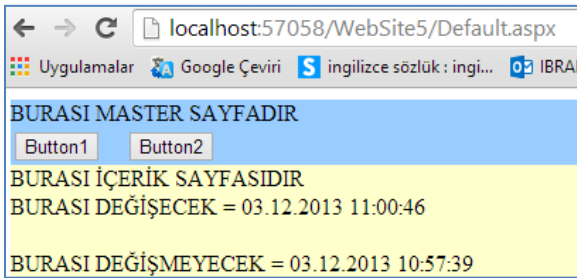
**NOT 2:** Bir önceki örnekte buton Update panelin içindeyken Ajax güncellemesi yapabiliyorduk. Ama her zaman butona Update panelin içerisinde olamaz. Sayfanın her hangi bir yerinde bulunmak zorunda olabilir. Bu durumda ne yapacağız?

Bu durumda Update panel ile Buton arasında bir köprü kurmalıyız. Bu işlem için Update Panelin Triggers (tetikleyici) ayarları yapılmalıdır. Burada hangi nesnelerin hangi olayda update paneli güncelleyeceği belirlenir.



### AJAX ÖRNEK 3:

MasterSayfadaki butona tıklanıldığında İçerik sayfasındaki belli bir bölgenin güncellenmesi.



#### Default.aspx (İçerik Sayfası)

```
<%@ Page Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default"
Title="Untitled Page" %>
```

```
<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
</asp:Content>
```

```
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
```

```
<div style="background-color: #FFFFCC">
```

**BURASI İÇERİK SAYFASIDIR**

```
<br />
```

```
<asp:UpdatePanel ID="UpdatePanel1" runat="server">
```

```
<ContentTemplate>
```

```

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

        </ContentTemplate>

        </asp:UpdatePanel>

        <br />

        BURASI DEĞİŞMEYECEK = <asp:Label ID="Label2" runat="server"
Text="Labe2"></asp:Label>

</div>

</asp:Content>

```

## Default.aspx.cs (İçerik Sayfası)

```

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        Label1.Text = DateTime.Now.ToString();
        Label2.Text = DateTime.Now.ToString();
    }
}

```

## Masterpage.master (Master Sayfa)

```

<%@ Master Language="C#" AutoEventWireup="true"
CodeFile="MasterPage.master.cs" Inherits="MasterPage" %>

<!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>
    <asp:ContentPlaceHolder id="head" runat="server">
    </asp:ContentPlaceHolder>
</head>
<body>
    <form id="form1" runat="server">

```







## DEFAULT.ASPX (İçerik Sayfası)

```
<%@ Page Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default"
Title="Untitled Page" %>
```

```
<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
</asp:Content>
```

```
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
```

```
Burasi Content (İçerik) sayfası
<br />
```

```
<asp:UpdatePanel id="UpdatePanel1" runat="server"
UpdateMode="Conditional">
```

```
<contenttemplate>
```

```
<fieldset>
```

```
<legend>UpdatePanel (Ajax'ın güncelleme paneli)</legend>
```

```
<br />
```

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

```
</fieldset>
```

```
</contenttemplate>
```

```
</asp:UpdatePanel>
```

```
</asp:Content>
```

## Default.aspx.cs

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
```

```

using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
}

```

## Masterpage.Master

```

<%@ Master Language="C#" AutoEventWireup="true"
CodeFile="MasterPage.master.cs" Inherits="MasterPage" %>

<!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>
    <asp:ContentPlaceHolder id="head" runat="server">
    </asp:ContentPlaceHolder>
</head>
<body>
    <form id="form1" runat="server">
    <div>

        Burası Master Sayfa
        <br />

        <asp:Button ID="Button2" runat="server" Text="Link 2"
            onclick="Button2_Click" />
        <asp:Button ID="Button1" runat="server" Text="Link 1"
            onclick="Button1_Click" />

        <br />

        <asp:ScriptManager ID="ScriptManager1" runat="server">
        </asp:ScriptManager>
        <br />

        <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">

        </asp:ContentPlaceHolder>
    </div>
    </form>
</body>
</html>

```

## Masterpage.Master.Cs

```

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;

```

```

using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Xml.Linq;

public partial class MasterPage : System.Web.UI.MasterPage
{
    protected void Page_Load(object sender, EventArgs e)
    {
        //Buton 1 ve 2 ile oluşturulacak ajax kodlarının senkronize
        çalışacağını belirliyor
        ScriptManager1.RegisterAsyncPostBackControl(Button1);
        ScriptManager1.RegisterAsyncPostBackControl(Button2);
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        string TiklananButonID = ((Control)sender).ID; //Butonun adını
        alıyor..
        LinkBagliGuncelleYap(TiklananButonID);
    }
    protected void Button2_Click(object sender, EventArgs e)
    {
        string TiklananButonID = ((Control)sender).ID;
        LinkBagliGuncelleYap(TiklananButonID);
    }

    public void LinkBagliGuncelleYap(string GelenID)
    {
        switch (GelenID)
        {
            case "Button1":
                ((Label)ContentPlaceHolder1.FindControl("Label1")).Text =
                "Link 1 e Tıklandı " + DateTime.Now.ToString();
                break;
            case "Button2":
                ((Label)ContentPlaceHolder1.FindControl("Label1")).Text =
                "Link 2 ye Tıklandı " + DateTime.Now.ToString();
                break;
        }

        ((UpdatePanel)ContentPlaceHolder1.FindControl("UpdatePanel1")).Update();
    }
}

```

NOT: Master Sayfadan İçerik sayfası Ajax ile güncellenirken şu 3 noktaya dikkat edin.


- 1) ScriptManager master sayfaya konulursa tüm içerik sayfaları için kullanabiliriz.
- 2) Master sayfadaki herhangi bir nesneye tıklanıldığında olayın gerçekleşmesini istiyorsak, Tıklayacağımız bu nesnelere ScriptManegar'a aşağıdaki kod ile MasterPagein load ında tanıtmalıyız.

```
ScriptManager1.RegisterAsyncPostBackControl(Button1);
```

- 3) Butona tıkladığımızda gerçekleşmesini istediğimiz kodları yazdıktan sonra hemen altında şu satırın olmasına dikkat edin.

```
((UpdatePanel)ContentPlaceHolder1.FindControl("UpdatePanel1")).Update();
```

- 4) Güncelleme işlemleri her zaman için değil özel durumlar için gerçekleşeceğini belirtmek için İçerik sayfasındaki Updatepanel seçilip onun **UpdateModu=Conditional** seçilmeli.

(ID)	<b>UpdatePanel1</b>
ChildrenAsTriggers	True
EnableViewState	True
RenderMode	Block
Triggers	(Collection)
UpdateMode	<b>Conditional</b> 
Visible	True

## AJAX ÖRNEK 5:

Master Sayfadan butonlara tıklayarak İçerik sayfası içindeki Update Panel'in güncellenmesinin yapılması. Panel içindeki takvimin günlerini artırıp azaltma işlemi.

**Master Page (Master Sayfa)**

---

---

**Content Page (İçerik Sayfası)**

UpdatePanel (Ajax'ın güncelleme paneli)

Kasım 2009						
<						>
Pzt	Sal	Çar	Per	Cum	Cmt	Paz
<u>26</u>	<u>27</u>	<u>28</u>	<u>29</u>	<u>30</u>	<u>31</u>	<u>1</u>
<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>
<u>9</u>	<u>10</u>	<u>11</u>	<u>12</u>	<u>13</u>	<u>14</u>	<u>15</u>
<u>16</u>	<u>17</u>	<u>18</u>	<u>19</u>	<u>20</u>	<u>21</u>	<u>22</u>
<u>23</u>	<u>24</u>	<u>25</u>	<u>26</u>	<u>27</u>	<u>28</u>	<u>29</u>
<u>30</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>

### MASTERPAGE.MASTER

```

<%@ Master Language="C#" AutoEventWireup="true"
CodeFile="MasterPage.master.cs" Inherits="MasterPage" %>

<!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 id="Head1" runat="server">
  <title>Master Sayfadan İçerik Sayfadaki Update Panel'i kullanma
</title>
</head>
<body>

  <form id="form1" runat="server">

  <div>
    Master Page (Master Sayfa)<br />
    <hr />
    <asp:Button ID="Button2" runat="server" Text="Eksi (-)"
      onclick="Button2_Click" />
    <asp:Button ID="Button1" runat="server" Text="Artı (+)"
      onclick="Button1_Click" />
    <hr />

    <asp:ScriptManager ID="ScriptManager1" runat="server">
    </asp:ScriptManager>
    <br />
  </div>
  </form>
  </body>
</html>

```

```
<asp:contentplaceholder id="ContentPlaceHolder1" runat="server">
</asp:contentplaceholder>

</div>

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

## MASTERPAGE.MASTER.CS

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Xml.Linq;

public partial class MasterPage : System.Web.UI.MasterPage
{
    public Int32 Hesapla
    {
        get
        { return (Int32)(ViewState["Hesapla"] ?? 0); }
        set
        { ViewState["Hesapla"] = value; }
    }

    protected void Page_Load(object sender, EventArgs e)
    {
        ScriptManager1.RegisterAsyncPostBackControl(Button1);
        ScriptManager1.RegisterAsyncPostBackControl(Button2);
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        switch (((Control)sender).ID)
        {
            case "Button1":
                this.Hesapla = this.Hesapla + 1;
                break;
            case "Button2":
                this.Hesapla = this.Hesapla - 1;
                break;
        }

        ((UpdatePanel)ContentPlaceHolder1.FindControl("UpdatePanell1")).Update();

        1Calendar cal = ((Calendar)ContentPlaceHolder1.FindControl("Calendar1"));
        DateTime newDateTime = DateTime.Today.Add(new TimeSpan(Hesapla, 0, 0, 0));
        cal.SelectedDate = newDateTime;
    }

    protected void Button2_Click(object sender, EventArgs e)
```

```

{
    switch (((Control) sender).ID)
    {
        case "Button1":
            this.Hesapla = this.Hesapla + 1;

            break;
        case "Button2":
            this.Hesapla = this.Hesapla - 1;
            break;
    }

    ((UpdatePanel)ContentPlaceholder1.FindControl("UpdatePanel1")).Update();
    Calendar cal =
    ((Calendar)ContentPlaceholder1.FindControl("Calendar1"));
    DateTime newDateTime = DateTime.Today.Add(new TimeSpan(Hesapla, 0,
0, 0));
    cal.SelectedDate = newDateTime;
}
}

```

\*\*\*\*\*

**Not:** public Int32 Hesapla { }

Bu kodlarla ilgili olarak şu yazmaktadır. ( Add the following code to create a public property named Offset in the master page that tracks the difference between today's date and the selected date. )

**Not:** Aşağıdaki kodlar için şu açıklama verilmiştir (Add the following code in the handler to register the two buttons as asynchronous postback controls:)

```

protected void Page_Load(object sender, EventArgs e)
{
    ScriptManager1.RegisterAsyncPostBackControl(Button1);
    ScriptManager1.RegisterAsyncPostBackControl(Button2);
}

```

## DEFAULT.ASPX

```

<%@ Page Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="Default2.aspx.cs" Inherits="Default2"
Title="Untitled Page" %>
<%@ MasterType VirtualPath="MasterPage.master" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">

    Content Page (İçerik Sayfası) <br />

    <asp:UpdatePanel id="UpdatePanel1" runat="server"
UpdateMode="Conditional">

        <contenttemplate>

            <fieldset>
                <legend>UpdatePanel (Ajax'ın güncelleme paneli)</legend>

                <br />
                <asp:Calendar id="Calendar1" runat="server"
onselectionchanged="Calendar1_SelectionChanged"></asp:Calendar>

```

```

        </fieldset>

        </contenttemplate>
    </asp:UpdatePanel>
</asp:Content>

```

## DEFAULT.ASPX.CS

```

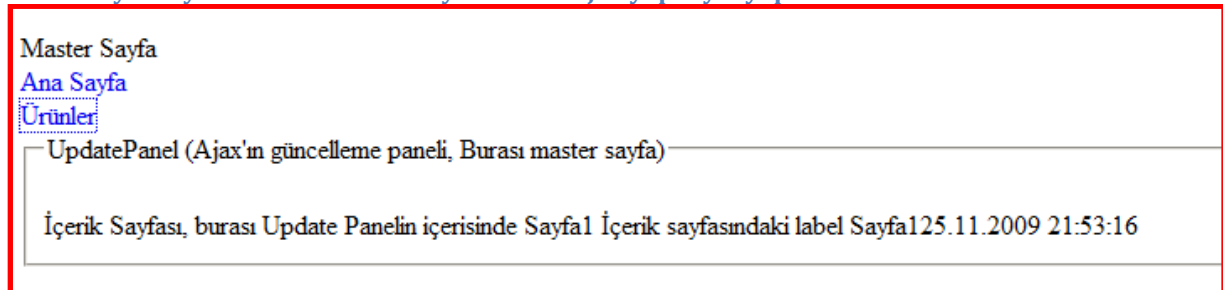
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class Default2 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        DateTime newDateTime =
            DateTime.Today.Add(new
                TimeSpan(Master.Hesapla, 0, 0, 0));
        Calendar1.SelectedDate = newDateTime;
    }
    protected void Calendar1_SelectionChanged(object sender, EventArgs e)
    {
        DateTime selectedDate = Calendar1.SelectedDate;
        Master.Hesapla
        = ((TimeSpan) Calendar1.SelectedDate.Subtract(DateTime.Today)).Days;
    }
}

```

## AJAX ÖRNEK 6: (Önemli) (HATALI ÇALIŞMIYOR)

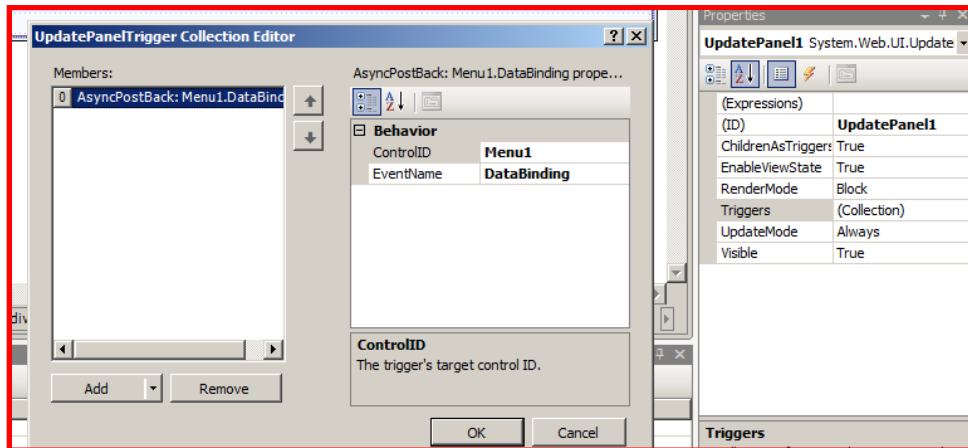
Master Sayfada Menu nesnesi bulunuyor. Menüye tıklayınca içerik sayfalarını ContentPlaceHolder kısmına yüklüyor. Fakat bu esnada yüklenme Ajax yapısıyla yapılmaktadır.



Bu işlem için kritik maddeler şunlardır.

- ScriptManager Master sayfaya konulacak
- Updatepanel Mastersayfanın içerisine konulmalı. Böylece içerik sayfaları Updatepanelin içerisinde açılmış olacaktır. Updatepanel ile Menü arasındaki Triggers işlemi aşağıdaki şekilde ayarlanmalıdır. Menü'nün Databinding olayı kullanılmalı.





Sayfalar çalışırken Menü'nün ID si ile ilgili bir problem meydana gelmektedir. Bu problemi aşabilmek için MasterSayfanın içerisine aşağıdaki kodlar eklenmeli

*(The reason for this message is that Controls in Content pages of Master pages do not actually have their original ID's. For example, although Menu1's ID is "Menu1" it's actual UniqueID is something like "ctl00\$Menu\$Menu1". The real problem is that you do not know this until runtime. You could add a AsyncPostBackTrigger() manually to the UpdatePanel.Triggers collection at runtime in the OnInit() function using static strings, but this is not very maintainable.*

*The following method simply looks through the triggers on the UpdatePanel and for the one that ends with the ID of the Menu1 control it simply changes the trigger ControlID to the UniqueID of Menu1 as runtime. This MUST happen in the OnInit() method of the Content Page. Although you have to hardcode the Menu1 item (and any other trigger controls), it means that you can change the Menu's ID or even the EventName without having to touch the code.)*

- `protected override void OnInit(EventArgs e)`
- `{`
- `base.OnInit(e);`
- `foreach (AsyncPostBackTrigger apbt in UpdatePanel1.Triggers)`
- `{`
- `if (Menu1.UniqueID.EndsWith(apbt.ControlID))`
- `apbt.ControlID = Menu1.UniqueID;`
- `}`
- `}`

### MasterPage.master

```
<%@ Master Language="C#" AutoEventWireup="true"
CodeFile="MasterPage.master.cs" Inherits="MasterPage" %>

<!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>MASTER SAYFA</title>
    <asp:ContentPlaceHolder id="head" runat="server">
    </asp:ContentPlaceHolder>
</head>
<body>
    <form id="form1" runat="server">
    <div>
        <br />
        <asp:ScriptManager ID="ScriptManager1" runat="server">
        </asp:ScriptManager>
    </div>
    </form>
</body>
</html>
```

```

Master Sayfa
<asp:Menu ID="Menu1" runat="server">
</asp:Menu>

<asp:UpdatePanel ID="UpdatePanel1" runat="server" >
  <ContentTemplate>

    <fieldset>
      <legend>UpdatePanel (Ajax'ın güncelleme paneli, Burası
master sayfa)</legend>
      <br />

      <asp:ContentPlaceHolder id="ContentPlaceHolder1"
runat="server">
        </asp:ContentPlaceHolder>

    </fieldset>
  </ContentTemplate>
  <Triggers>
    <asp:AsyncPostBackTrigger ControlID="Menu1"
EventName="DataBinding" />
  </Triggers>
</asp:UpdatePanel>

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

```

#### MasterPage.master.cs

```

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Xml.Linq;

public partial class MasterPage : System.Web.UI.MasterPage
{
  protected void Page_Load(object sender, EventArgs e)
  {
    if (!IsPostBack)
    {
      MenuItem Eleman = new MenuItem();
      Eleman = new MenuItem();
      Eleman.Text = "Ana Sayfa";
      Eleman.Value = "AnaSayfa";
      Menu1.Items.Add(Eleman);

      Eleman = new MenuItem();
      Eleman.Text = "Ürünler";
      Eleman.Value = "Urunler";
      Menu1.Items.Add(Eleman);
    }
  }
}

```

```

    }
}
protected void Menu1_MenuItemClick(object sender, MenuEventArgs e)
{
    switch (e.Item.Value)
    {
        case "AnaSayfa":
            e.Item.NavigateUrl = "Default.aspx";
            return;

        case "Urunler":
            e.Item.NavigateUrl = "Default2.aspx";
            return;
    }
}

protected override void OnInit(EventArgs e)
{
    base.OnInit(e);
    foreach (AsyncPostBackTrigger apbt in UpdatePanel1.Triggers)
    {
        if (Menu1.UniqueID.EndsWith(apbt.ControlID))
            apbt.ControlID = Menu1.UniqueID;
    }
}
}

```

#### Default.aspx

```

<%@ Page Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default"
Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    İçerik Sayfası, burası Update Panelin içerisinde Sayfal

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

</asp:Content>

```

#### Default.aspx.cs

```

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class _Default : System.Web.UI.Page

```

```
{
    protected void Page_Load(object sender, EventArgs e)
    {
        Label1.Text = "İçerik sayfasındaki label Sayfa1" + DateTime.Now;
    }
}
```

### Default2.aspx

```
<%@ Page Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="Default2.aspx.cs" Inherits="Default2"
Title="Untitled Page" %>
```

```
<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
```

İçerik Sayfası, burası Update Panelin içerisinde Sayfa2

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

```
</asp:Content>
```

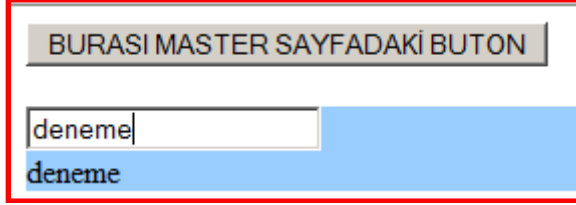
### Default2.aspx.cs

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
```

```
public partial class Default2 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        Label1.Text = "İçerik sayfasındaki label Sayfa2" + DateTime.Now;
    }
}
```

## MASTER SAYFADAN BİR BUTONA TIKLAYARAK İÇERİK SAYFASINDAKİ BİLGİLERİN DEĞİŞTİRİLMESİ (Ajax'la ilgili değil)

### ÖRNEK-7



#### MasterPage.master

```
<%@ Master Language="C#" AutoEventWireup="true"
CodeFile="MasterPage.master.cs" Inherits="MasterPage" %>

<!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>
  <asp:ContentPlaceHolder id="head" runat="server">
  </asp:ContentPlaceHolder>
</head>
<body>
  <form id="form1" runat="server">
    <div>

      <asp:Button ID="Button1" runat="server" Text="BURASI MASTER
SAYFADAKİ BUTON"
        onclick="Button1_Click" />
      <br />
      <br />

      <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">

      </asp:ContentPlaceHolder>
    </div>
  </form>
</body>
</html>
```

#### MasterPage.master.cs

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Xml.Linq;
```

```

public partial class MasterPage : System.Web.UI.MasterPage
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void Button1_Click(object sender, EventArgs e)
    {

    ((Label)ContentPlaceHolder1.FindControl("Label1")).Text=((TextBox)ContentPl
aceHolder1.FindControl("Textbox1")).Text;

    }
}

```

### Default.aspx

```

<%@ Page Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default"
Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <title> BURASI ICERIK SAYFASININ BAŞLIĞI</title>

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">

    <div style="background-color: #99CCFF">
        <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>

        <br />

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

    </div>

</asp:Content>

```

### Default.aspx.cs

```

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
}

```