

UPDATE PANEL MANTIĞI VE KULLANIMI

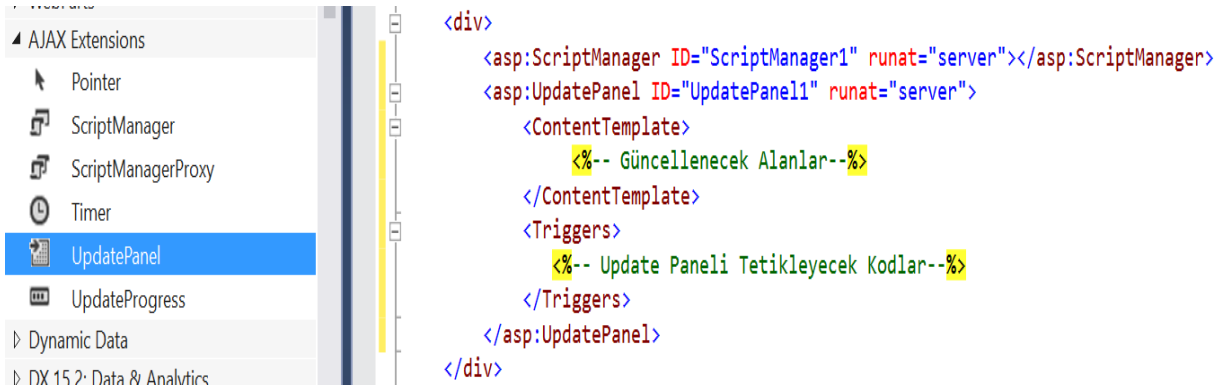
Bu yazımızda Asp.Net Ajax uygulamalarında kullanılan Update Panel kontrolünün ne işe yaradığını, neden ve nasıl kullanıldığını anlatmaya çalışıyor olacağız.

Update Panel web sayfalarında belli kısımlar, bölümler oluşturup bu bölümlerin Postback olmadan sayfadan bağımsız bir şekilde Asenkron olarak güncellenmesine yarayan Asp.Net kontrolüdür. Sayfamızda bir çok nesne ve veri olduğunu düşünürsek, bazen tüm sayfanın server-client arasında Postback ile gidip gelmesi hem yavaşlığa hem sistem performans sorunlarına sebep olabilir. Güncellenmesini istediğimiz küçük bir alan (Textbox,Grid vb) için tüm sayfanın gereksiz yere Postback olması genelde istemediğimiz bir durumdur. Bu tip durumlara Update Panel kullanılırsa Server-Client arasındaki gereksiz veri trafiği engellenmiş ve sadece Update Panel içindeki alanlar asenkron olarak güncellenmiş olur.

Update Panel kontrolü;

```
<ContentTemplate> </ContentTemplate>
<Triggers> </Triggers>
```

adında 2 kısımdan oluşmaktadır. Update Paneli kullanabilmemiz için sayfamıza ScriptManager'i eklememiz gerekmektedir. ScriptManager, AJAX kontrollerini kullanırken mutlaka eklememiz gereken yönetici görevindeki kontroldür ve en üstte bulunması gerekmektedir. ScriptManager sayfaya ilgili AJAX kontrolleri için scriptlerin yüklenmesini ve yönetilmesini sağlar. Hem ScriptManager'i hem de UpdatePanel'i Toolbox -> AJAX Extensions altında bulabilirsiniz.



ContentTemplate(İçerik Şablonu):

ContentTemplate içerisinde Asp.Net ve Html kontrolleri yer almaktadır. Bu alana Update Panelde güncellenmesini istediğimiz alanları eklemekteyiz. Web sitesinde yer alabilecek her türlü nesne, resim, yazı, metin bu alana eklenebilir. Update Panel kullanılırken sadece bu alan içerisindeki alanlar asenkron olarak güncellenecektir.

Triggers:

Bu alanda ContentTemplate içeriğinin UpdatePanel dışındaki herhangi bir nesneye ait hangi olay veya olaylarla tetiklenip update edileceği belirtilir. Örnek verecek olursak, UpdatePanel dışına konulan bir butonun "click" eventi buraya eklenirse, butona her tıklandığında sayfa PostBack olmadan sadece UpdatePanel içi güncellenir. Ya da herhangi bir Dropdown'ın SelectedChange eventü UpdatePanel'e bağlanabilir. Trigger içine yazılabilecek 2 kavram mevcuttur.

"AsyncPostBackTrigger" : Bu seçenekle eklenen event sadece asenkron olarak PostBack olmadan UpdatePanel içerisindeki güncellenmesini sağlar.(Default)

"PostBackTrigger" : Bu seçenekle eklenen event ise, sadece UpdatePanel içerisinin değil, PostBack yaparak tüm sayfanın güncellenmesini sağlar. UpdatePanel içerisine konulan tetikleyici bir kontrolün tüm sayfayı PostBack yapması istenildiği durumlarda kullanılabilir.

UpdatePanel'in sayfadan asenkron olarak güncellenebilmesi için tetiklenmesi gerekmektedir. Bu tetikleme işlemi temel olarak tetikleyici kontrolün (Button, Dropdown, ComboBox vs) yerine bağlıdır. Durumlar bu kontrolün UpdatePanel'in içinde veya dışında olmasına göre değişecektir.

Örnek -1 :

İlk örneğimizde UpdatePanel kullanılmadığı zaman sayfanın tümünün her butona tıklandığında PostBack olarak yeniden yüklendiğini göreceğiz.

Aspx:

```
<body>
  <form id="form1" runat="server">
    <div style="border: 1px dotted red; padding: 5px; margin: 5px; float: left;">
      <div id="sonuc" runat="server"></div>
      <br />
      <asp:TextBox ID="Textbox1" runat="server"></asp:TextBox><br />
      <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Button" />
    </div>
    <br />
    <div id="divExt" runat="server"></div>
  </form>
</body>
```

Code Behind:

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

0 references
protected void Button1_Click(object sender, EventArgs e)
{
  sonuc.InnerHtml += "<br />" + Textbox1.Text;
  divExt.InnerHtml += "<br />" + Textbox1.Text + " - Update Panel Dışı";
  Textbox1.Text = null;
  Textbox1.Focus();
}
```

Çıktısı:

Test1
Test2
TestN

Button

Test1 - Update Panel Dışı
Test2 - Update Panel Dışı
TestN - Update Panel Dışı

Görüldüğü üzere sayfada UpdatePanel kullanılmadığı için, textbox alanına herhangi bir yazı girilip her butona tıklandığında kırmızı çizgili alanın (burası sonraki örnekte UpdatePanel içine alınacak) içi ve dışı PostBack ile aynı anda refresh olacaktır. Sayfanın, butona her basıldığında PostBack olduğunu üst kısımdaki "refresh" ifadesinden de takip edebilirsiniz.

Örnek -2 (Tetikleyici Kontrol UpdatePanel İçinde):

Şimdi aynı örnekteki kırmızı çizgili alanımızı UpdatePanel içine alalım. Böylelikle butona her tıklandığında yalnızca kırmızı çizgili alanın güncellendiğini, sağ tarafındaki UpdatePanel dışında kalan alanın safya PostBack olmadığı için güncellenmediğini izleyelim. CodeBehind tarafında hiç bir değişiklik yapmadan sadece Aspx tarafında aşağıdaki değişikliği yapalım.

Aspx:

```
<body>
  <form id="form1" runat="server">
    <asp:ScriptManager ID="Scriptmanager1" runat="server"></asp:ScriptManager>
    <div style="border: 1px dotted red; padding: 5px; margin: 5px; float: left;">
      <asp:UpdatePanel ID="UpdatePanel1" runat="server">
        <ContentTemplate>
          <div id="sonuc" runat="server"></div>
          <br />
          <asp:TextBox ID="Textbox1" runat="server"></asp:TextBox><br />
          <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Button" />
        </ContentTemplate>
        <Triggers> </Triggers>
      </asp:UpdatePanel>
    </div>
    <br />
    <div id="divExt" runat="server"></div>
  </form>
</body>
```

Çıktısı:



Görüldüğü gibi text alana girilen yazıdan sonra butona her basıldığında sadece UpdatePanel içerisi güncellenmiş, UpdatePanel dışında kalan alanlar sayfa PostBack olmadığı için güncellenmemiştir. CodeBehind kısmında herhangi bir değişiklik yapmadık ve sağ taraftaki div içeriği update olmadı. Bu bize sayfanın PostBack olmadığını, UpdatePanelin doğru çalıştığını gösterir. Konuyu daha iyi kavramak adına UpdatePanel dışına bir buton koyup, bu butonun eventinde sağ taraftaki div içeriğini olduğu gibi yazdırmayı denerseniz, o zaman yeni eklediğiniz butona her tıkladığınızda sayfanın komple PostBack olduğunu görür ve sağ taraftaki div içeriğinin update olduğunu görebilirsiniz.

Soru ?

UpdatePanel içerisindeki "Trigger" alanını boş bırakmamıza rağmen UpdatePanel nasıl oldu da tetiklendi? Eğer "Trigger" alanı boş bırakılarak da UpdatePanel tetiklenebiliyorsa bu alan neden var?

UpdatePanelin **"ChildrenAsTrigger"** özelliğinin default değeri "true" olduğu için (**"UpdateMode"** özelliğinin de default olarak "Always"), UpdatePaneli tetikleyecek kontrol eğer UpdatePanel içerisinde yer alıyorsa herhangi bir trigger yazmadan otomatik olarak UpdatePaneli tetikleyecektir. Eğer bu değer "false" yapılırsa, o zaman UpdatePaneli tetikleyecek olan kontrol, UpdatePanel içinde dahi olsa otomatik olarak tetiklenmez ve trigger alanında belirtmemiz gerekecekti. Tabi **"ChildrenAsTrigger"** özelliğinin "false" olabilmesi için **"UpdateMode"** özelliğinin **"Conditional"** a çekilmesi gerekmektedir.

Örnek -3 :

```
<asp:ScriptManager ID="Scriptmanager1" runat="server"></asp:ScriptManager>
<div style="border: 1px dotted red; padding: 5px; margin: 5px; float: left;">
  <asp:UpdatePanel ID="UpdatePanel1" runat="server" UpdateMode="Conditional" ChildrenAsTriggers="false">
    <ContentTemplate>
      <div id="sonuc" runat="server"></div>
      <br />
      <asp:TextBox ID="Textbox1" runat="server"></asp:TextBox><br />
      <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Button" />
    </ContentTemplate>
    <Triggers> </Triggers>
  </asp:UpdatePanel>
</div>
<br />
<div id="divExt" runat="server"></div>
```

Yukarıdaki gibi tanımlamalar yapılırsa, butona istediğiniz kadar tıklayın UpdatePanel güncellenmeyecektir.

Örnek -4 :

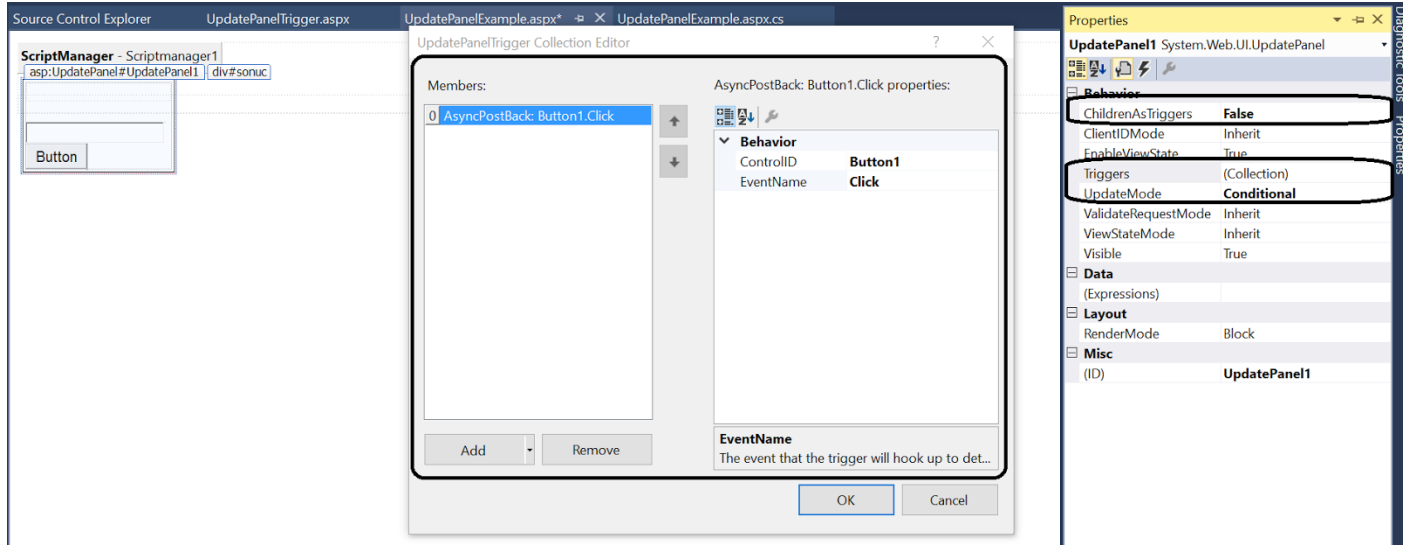
UpdatePanelin güncellenebilmesi için aşağıdaki gibi Trigger tanımlanması gerekecektir. Bu tanımlamadan sonra butona basıldığında UpdatePanel tetiklenecek ve içeriği asenkron olarak güncellenecektir.

```

<asp:ScriptManager ID="Scriptmanager1" runat="server"></asp:ScriptManager>
<div style="border: 1px dotted red; padding: 5px; margin: 5px; float: left;">
  <asp:UpdatePanel ID="UpdatePanel1" runat="server" UpdateMode="Conditional" ChildrenAsTriggers="false">
    <ContentTemplate>
      <div id="sonuc" runat="server"></div>
      <br />
      <asp:TextBox ID="Textbox1" runat="server"></asp:TextBox><br />
      <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Button" />
    </ContentTemplate>
    <Triggers>
      <asp:AsyncPostBackTrigger ControlID="Button1" EventName="Click" />
    </Triggers>
  </asp:UpdatePanel>
</div>
<br />
<div id="divExt" runat="server"></div>

```

UpdatePanelin Trigger ve diğer özelliklerini Aspx sayfasında direk olarak yazabileceğiniz gibi, dizayn moda geçip "Properties" üzerinden de yapabilirsiniz, Aspx sayfasına döndüğünüzde ilgili kodların otomatik olarak yaratıldığını göreceksiniz.



Örnek -5 (Tetikleyici Kontrol UpdatePanel Dışında):

Tetikleyici kontrol nesnenin UpdatePanel içerisinde olması durumunda default ayarlarda tetikleme yapması için herhangi bir Trigger yazmamız gerekmediğinden bahsettik. Fakat tetikleyici kontrol UpdatePanel dışında bir yerde ise ve biz bu kontrolle UpdatePaneli tetiklemek istiyorsak "ChildrenAsTrigger" özelliğine bakmaksızın Trigger tanımlaması yapmak zorundayız. Eğer ilgili kontrolü UpdatePanele bağlamazsak yani Trigger tanımlaması yapmazsak, tüm sayfanın PostBack olacağını unutmayalım.

Şimdi ekranımıza bir UpdatePanel ekleyip, içine sistem tarih ve saatini basıp, ayrıca butona her basıldığında sırayla açılıp kapanacak 2 panel ekleyelim. Bunlardan sonra da UpdatePanel dışına sistem tarih saati basıp ayrıca 2 adet buton ekleyip, bir tanesini Trigger olarak UpdatePanele bağlayalım.

Aspx:

```

<asp:ScriptManager ID="ScriptManager1" runat="server"></asp:ScriptManager>
<div style="border: 1px dotted red; padding: 5px; margin: 5px; width:200px; ">
  <asp:UpdatePanel ID="UpdatePanel1" runat="server">
    <ContentTemplate>
      <%=DateTime.Now %>
      <asp:Panel ID="Panel1" runat="server">
        <p>Panel 1 Açık</p>
      </asp:Panel>
      <asp:Panel ID="Panel2" runat="server">
        <p>Panel 2 Açık</p>
      </asp:Panel>
    </ContentTemplate>

    <Triggers>
      <asp:AsyncPostBackTrigger ControlID="Button1" EventName="Click" />
    </Triggers>

  </asp:UpdatePanel>
</div>
<%=DateTime.Now %>
<br />
<asp:Button ID="Button1" runat="server" Text="UpPanel Refresh" OnClick="Button1_Click" />
<asp:Button ID="Button2" runat="server" Text="Page Postback" OnClick="Button2_Click" />

```

CodeBehind:

0 references

```

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

```

0 references

```

protected void Button2_Click(object sender, EventArgs e)
{
    PanelVisibleChange();
}

```

2 references

```

private void PanelVisibleChange()
{
    if (Panel1.Visible == false)
        Panel1.Visible = true;
    else
    {
        Panel1.Visible = false;
        Panel2.Visible = true;
    }
}

```

Çıktısı:

9.03.2016 16:40:41

Panel 1 Açık

9.03.2016 16:39:30

UpPanel Refresh

Page Postback

Sonuçta görüldüğü gibi "Button1" kontrolünün "Click" eventini UpdatePanele bağladığımız için, bu butona basılınca panel içerisi PostBack olmadan asenkron olarak güncellendi (saatlerin farklı olduğuna dikkat ediniz, ilk açılışta saatler aynıydı). Button2'ye tıklandığında, bu butonun UpdatePanel ile herhangi bir ilgisi olmadığı için sayfa komple PostBack olur ve hem UpdatePanel içi, hemde UpdatePanel dışı aynı anda senkron bir şekilde güncellenir.

Birden Fazla Update Panel Kullanımı (Önemli Nokta):

Bir sayfada birden fazla UpdatePanel olabilir. UpdatePanel tanımlamalarında herhangi bir sıkıntı ya da değişiklik olmaz. Burada dikkat edilmesi gereken nokta örneğin A ve B adında 2 UpdatePanel olduğunu düşünürsek, birini tetikleyen olayın diğerini de tetikleyeceği gerçeğidir. Aslında A UpdatePanelinin trigger olayının, B panelini tetiklememesi gerekir diye düşünülebilir, bunun sebebi UpdatePanellerin "UpdateMode" özelliklerinin "Always" olmasıdır. Örneğin A UpdatePanelin "UpdateMode" özelliği "Conditional" olarak değiştirilirse, B UpdatePanel'in tetikleyicisinden etkilenmeyecektir. Dolayısıyla A paneli sadece kendi trigger tanımında yer alan kontrol tarafından tetiklenecektir, fakat B paneli halen "Always" olduğundan A panelin kontrolü tarafından da tetiklenecektir. "UpdateMode" özelliği "Always" olan UpdatePaneller sayfada yer alan herhangi bir UpdatePanel tetiklendiği zaman otomatik olarak tetiklenecektir.

İç İçe Update Panel Kullanımı:

Bazı durumlarda iç içe UpdatePanel kullanılmak istenebilir. Örneğin ekranın tamamını kaplayan bir UpdatePanel düşünülürse, bu UpdatePanelin içine 1 veya 1'den fazla UpdatePanellerde gömülebilir. Eklenen bu UpdatePanel'lerinde birbirinden farklı asenkron olarak güncellenmesi istenebilir. Bu durumda yine "UpdateMode" özelliğinden faydalananarak iç içe geçmiş UpdatePanel yapılarının birbirinden bağımsız tetiklenmesini sağlıyor olacağız.

Örnek -6 :

Şimdiki örneğimizde dış kısma bir master UpdatePanel ekleyip içerisine sistem saatini yazdıralım ve bir de otomatik tetikleyici buton ekleyelim. Daha sonra master UpdatePanelin ContentTemplate kısmına bir tanede child UpdatePanel ekleyelim, aynı işlemleri bu panele de uygulayalım. Aşağıdaki koddan dikkat edersiniz her iki panelin de "UpdateMode" özelliği default "Always" olduğu için herhangi birinin tetikleyicisi diğerini de tetikler ve ekrandaki sistem saati aynı anda güncellenir.

Aspx:

```

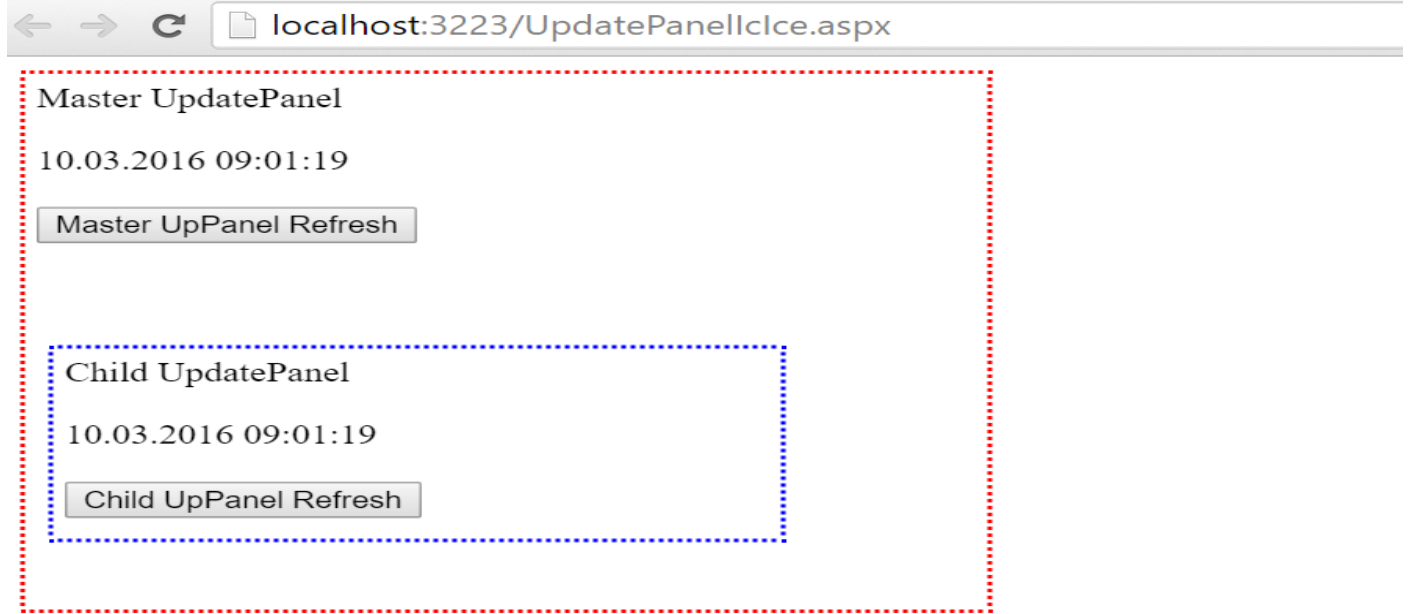
<asp:ScriptManager ID="ScriptManager1" runat="server"></asp:ScriptManager>
<div style="border: 2px dotted red; padding: 5px; margin: 5px; width: 400px; height: 300px;">
  <asp:UpdatePanel ID="UpdatePanel1" runat="server">
    <ContentTemplate>
      Master UpdatePanel </br></br>
      <%=DateTime.Now %> </br></br>
      <asp:Button ID="Button1" runat="server" Text="Master UpPanel Refresh" />

      </br></br></br></br>

      <div style="border: 2px dotted blue; padding: 5px; margin: 5px; width: 300px; height: 100px;">
        <asp:UpdatePanel ID="UpdatePanel2" runat="server">
          <ContentTemplate>
            Child UpdatePanel </br></br>
            <%=DateTime.Now %> </br></br>
            <asp:Button ID="Button2" runat="server" Text="Child UpPanel Refresh" />
          </ContentTemplate>
          <Triggers></Triggers>
        </asp:UpdatePanel>
      </div>
    </ContentTemplate>
    <Triggers></Triggers>
  </asp:UpdatePanel>
</div>

```

Çıktısı:



Sonuçta da görüldüğü gibi hangi butona tıklanırsa tıklansın, UpdatePanellerin ikisi de aynı olaydan tetikleniyor. Sistem saatinin aynı olmasından anlayabilirsiniz.

Örnek -7 :

Şimdi child UpdatePanel tetikleyicisini Master panelin tetikleyicinden ayırmak istiyorum, bunun için tek yapmam gereken şey master panelin "UpdateMode" özelliğini değiştirip "Conditional" yapmaktır. Böylelikle master paneli sadece kendi tetikleyicisine bağlamış oluyorum, child UpdatePanel refresh butonuna tıkladığımda sadece child panel içeriğinin güncellendiğini, master panelin güncellenmediğini görebilirim.

Aspx:

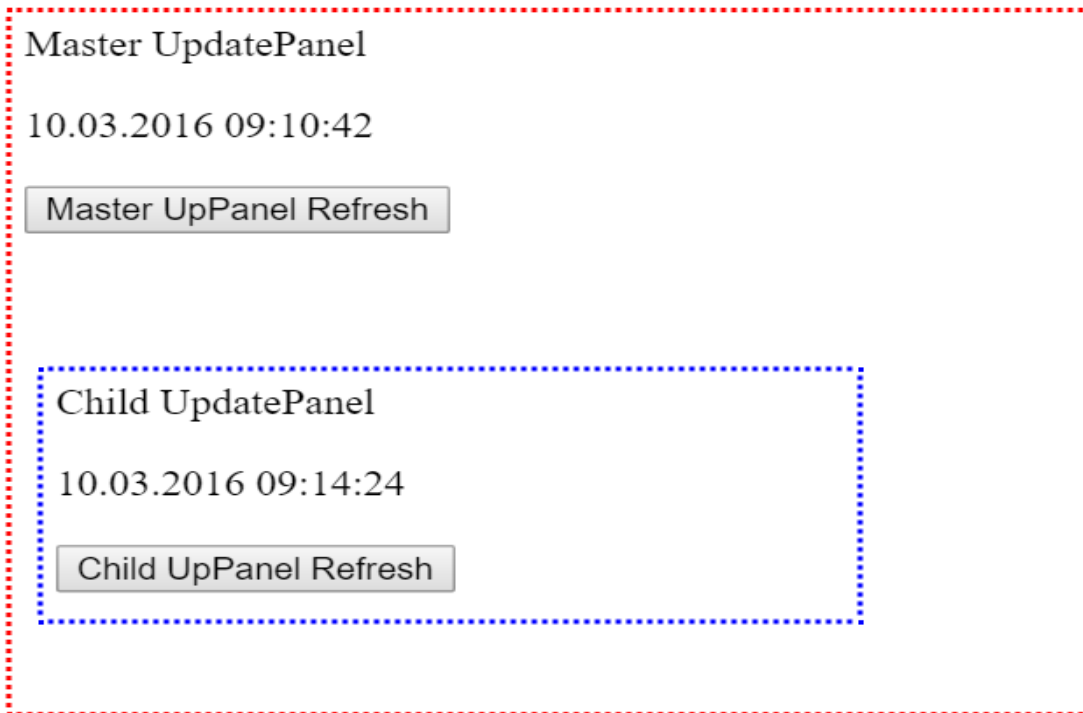

```

<div style="border: 2px dotted red; padding: 5px; margin: 5px; width: 400px; height: 300px;"
  <asp:UpdatePanel ID="UpdatePanel1" runat="server" UpdateMode="Conditional">
    <ContentTemplate>
      Master UpdatePanel </br></br>
      <%=DateTime.Now %> </br></br>
      <asp:Button ID="Button1" runat="server" Text="Master UpPanel Refresh" />
    </ContentTemplate>
  </asp:UpdatePanel>

  <div style="border: 2px dotted blue; padding: 5px; margin: 5px; width: 300px; height: 100px;"
    <asp:UpdatePanel ID="UpdatePanel2" runat="server" >
      <ContentTemplate>
        Child UpdatePanel </br></br>
        <%=DateTime.Now %> </br></br>
        <asp:Button ID="Button2" runat="server" Text="Child UpPanel Refresh" />
      </ContentTemplate>
      <Triggers></Triggers>
    </asp:UpdatePanel>
  </div>
</ContentTemplate>
<Triggers></Triggers>
</asp:UpdatePanel>
</div>

```

localhost:3223/UpdatePanelIce.aspx



Not:

İç kısımda yer alan child UpdatePanelin "UpdateMode" özelliğini değiştirip "Conditional" yapmanız herhangi bir şey değiştirmeyecektir. İçinde bulunduğu Master UpdatePanel'in tetikleyicisinden her hâlükârda otomatik olarak tetiklenecektir. Dolayısıyla master panelde yer alan buton her durumda içinde bulunan tüm UpdatePanelleri tetikleyecektir.

Örnek -8 :

Konunun daha net anlaşılması için bir alt kısma bir panel daha ekleyip, tüm panellerin "UpdateMode" özelliğini değiştirip "Conditional" yapalım. Bu durumda master panele ait buton iç kısımdaki tüm panelleri tetikleyecek, fakat

child1 ve child2 paneli içindeki butonlar sadece kendi UpdatePanellerini tetikleyip, diğer panellerden bağımsız çalışacaklardır.

Aspx:

```
<div style="border: 2px dotted red; padding: 5px; margin: 5px; width: 400px; height: 400px;">
  <asp:UpdatePanel ID="UpdatePanel1" runat="server" UpdateMode="Conditional">
    <ContentTemplate>
      Master UpdatePanel </br></br>
      <%=DateTime.Now %> </br></br>
      <asp:Button ID="Button1" runat="server" Text="Master UpPanel Refresh" />
    </ContentTemplate>
  </asp:UpdatePanel>

  <div style="border: 2px dotted blue; padding: 5px; margin: 5px; width: 300px; height: 100px;">
    <asp:UpdatePanel ID="UpdatePanel2" runat="server" UpdateMode="Conditional" >
      <ContentTemplate>
        Child1 UpdatePanel </br></br>
        <%=DateTime.Now %> </br></br>
        <asp:Button ID="Button2" runat="server" Text="Child1 UpPanel Refresh" />
      </ContentTemplate>
      <Triggers></Triggers>
    </asp:UpdatePanel>
  </div>

  <div style="border: 2px dotted green; padding: 5px; margin: 5px; width: 300px; height: 100px;">
    <asp:UpdatePanel ID="UpdatePanel3" runat="server" UpdateMode="Conditional">
      <ContentTemplate>
        Child2 UpdatePanel </br></br>
        <%=DateTime.Now %> </br></br>
        <asp:Button ID="Button3" runat="server" Text="Child2 UpPanel Refresh" />
      </ContentTemplate>
      <Triggers></Triggers>
    </asp:UpdatePanel>
  </div>
</div>
```



Master UpdatePanel

10.03.2016 09:36:50

Master UpPanel Refresh

Child1 UpdatePanel

10.03.2016 09:37:25

Child1 UpPanel Refresh

Child2 UpdatePanel

10.03.2016 09:44:21

Child2 UpPanel Refresh