عنوان: **تبدیل پلاگین‌های jQuery به کنترل‌های ASP.Net**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۰۹/۰۸ ۱۵:۵۸:۱۸
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net

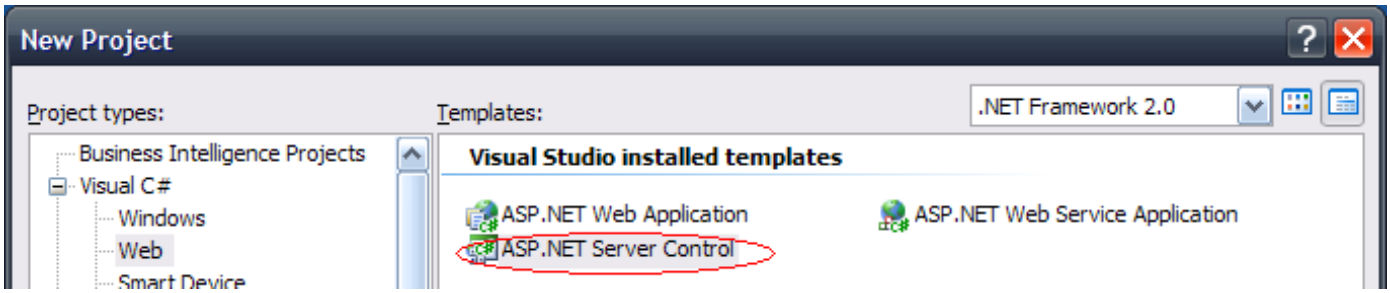
امروز داشتم یک سری از پلاگین‌های jQuery را مرور می‌کردم، مورد زیر به نظرم واقعا حرفه‌ای اومد و کمبود آن هم در بین کنترل‌های استاندارد ASP.Net محسوس است:

Masked Input Plugin

استفاده از آن به صورت معمولی بسیار ساده است. فقط کافی است اسکریپت‌های jQuery و سپس این افزونه به هدر صفحه اضافه شوند و بعد هم مطابق صفحه usage آن عمل کرد.

خیلی هم عالی! ولی این شیوه‌ی متداول کار در ASP.Net نیست. آیا بهتر نیست این مجموعه را تبدیل به یک کنترل کنیم و از این پس به سادگی با استفاده از Toolbox ویژوال استودیو آن‌را به صفحات اضافه کرده و بدون درگیر شدن با دستکاری سورس html صفحه، از آن استفاده کنیم؟ به‌عبارتی دیگر یکبار باید با جزئیات درگیر شد، آنرا بسته بندی کرد و سپس بارها از آن استفاده نمود. (مفاهیم شیءگرایی)

برای این‌کار، یک پروژه جدید ایجاد ASP.Net server control را آغاز نمائید (به نام MaskedEditCtrl).



به صورت پیش فرض یک قالب استاندارد ایجاد خواهد شد که کمی نیاز به اصلاح دارد. نام کلاس را به MaskedEdit تغییر خواهیم داد و همچنین در قسمت ToolboxData نیز نام کنترل را به MaskedEdit ویرایش می‌کنیم.

برای اینکه مجبور نشویم یک کنترل کاملا جدید را از صفر ایجاد کنیم، خواص و توانایی‌های اصلی این کنترل را از TextBox استاندارد به ارث خواهیم برد. بنابراین تا اینجای کار داریم:

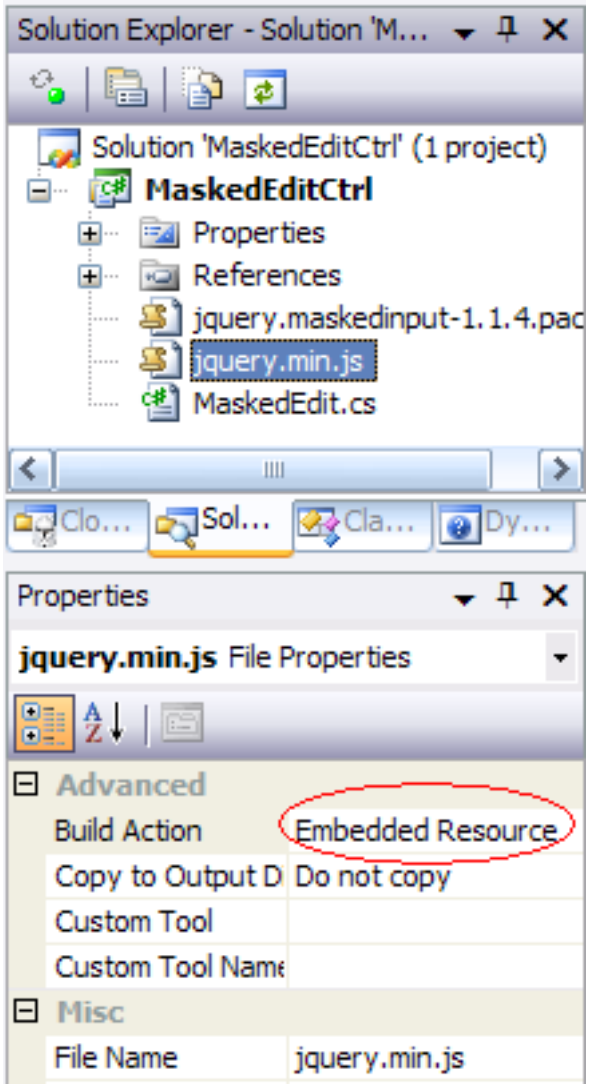
```
namespace MaskedEditCtrl
{
   [DefaultProperty("MaskFormula")]
   [ToolboxData("<{0}:MaskedEdit runat=server></{0}:MaskedEdit>")]
   [Description("MaskedEdit Control")]
   public class MaskedEdit : TextBox

   {
```

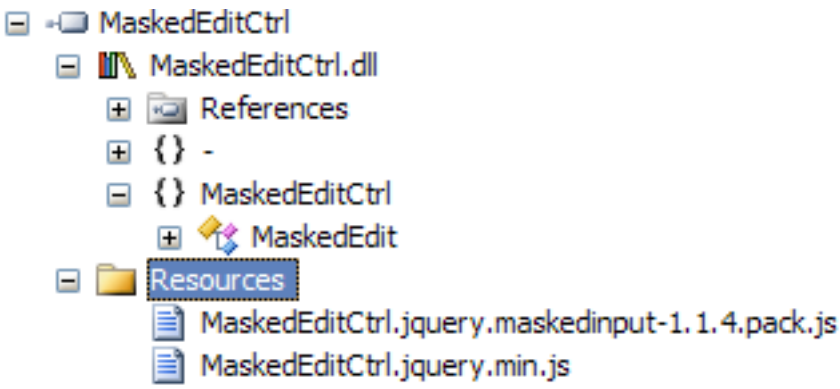
سپس باید رویداد OnPreRender را تحریف (override) کرده و در آن همان اعمالی را که هنگام افزودن اسکریپت‌ها به صورت دستی انجام می‌دادیم با برنامه نویسی پیاده سازی کنیم. چند نکته ریز در اینجا وجود دارد که در ادامه به آن‌ها اشاره خواهد شد.

از ASP.Net 2.0 به بعد، امکان قرار دادن فایل‌های اسکریپت و یا تصاویر همراه یک کنترل، درون فایل dll آن بدون نیاز به توزیع مجزای آنها به صورت WebResource مهیا شده است. برای این منظور اسکریپت‌های jQuery و افزونه mask edit را به پروژه اضافه

نمائید. سپس به قسمت خواص آنها (هر دو اسکریپت) مراجعه کرده و build action آنها را به Embedded Resource تغییر دهید (شکل زیر):



از این پس با کامپایل پروژه، این فایل‌ها به صورت resource به dll ما اضافه خواهند شد. برای تست این مورد می‌توان به برنامه reflector مراجعه کرد (تصویر زیر):



پس از افزودن مقدماتی اسکریپت‌ها و تعریف آنها به صورت resource ، باید آنها را در فایل AssemblyInfo.cs پروژه نیز تعریف کرد (به صورت زیر).

```
[assembly: WebResource("MaskedEditCtrl.jquery.min.js", "text/javascript")]
[assembly: WebResource("MaskedEditCtrl.jquery.maskedinput-1.1.4.pack.js", "text/javascript")]
```

نکته مهم: همانطور که ملاحظه می‌کنید نام فضای نام پروژه (namespace) باید به ابتدای اسکریپت‌های معرفی شده اضافه شود.

پس از آن نوبت به افزودن این اسکریپت‌ها به صورت خودکار در هنگام نمایش کنترل است. برای این منظور داریم:

```
protected override void OnPreRender(EventArgs e)
       {
           base.OnPreRender(e);

           //adding .js files
           if (!Page.ClientScript.IsClientScriptIncludeRegistered("jquery_base"))
           {
               string scriptUrl = Page.ClientScript.GetWebResourceUrl(this.GetType(),
                   "MaskedEditCtrl.jquery.min.js");
               Page.ClientScript.RegisterClientScriptInclude("jquery_base", scriptUrl);
           }

           if (!Page.ClientScript.IsClientScriptIncludeRegistered("edit_ctrl"))
           {
               string scriptUrl = Page.ClientScript.GetWebResourceUrl(this.GetType(),
                   "MaskedEditCtrl.jquery.maskedinput-1.1.4.pack.js");
               Page.ClientScript.RegisterClientScriptInclude("edit_ctrl", scriptUrl);
           }

           if (!Page.ClientScript.IsStartupScriptRegistered("MaskStartup" + this.ID))
           {
               // Form the script to be registered at client side.
               StringBuilder sbStartupScript = new StringBuilder();
               sbStartupScript.AppendLine("jQuery(function($){");
               sbStartupScript.AppendLine("$(\"#" + this.ClientID + "\").mask(\"" + MaskFormula +
"\");");
               sbStartupScript.AppendLine("});");
               Page.ClientScript.RegisterStartupScript(typeof(Page),
                   "MaskStartup" + this.ID, sbStartupScript.ToString(), true);
           }

       }
```

همانطور که ملاحظه می‌کنید، ابتدا WebResource دریافت شده و سپس به صفحه اضافه می‌شود. در آخر مطابق راهنمای افزونه mask edit عمل شد. یعنی اسکریپت مورد نظر را ساخته و به صفحه اضافه کردیم.

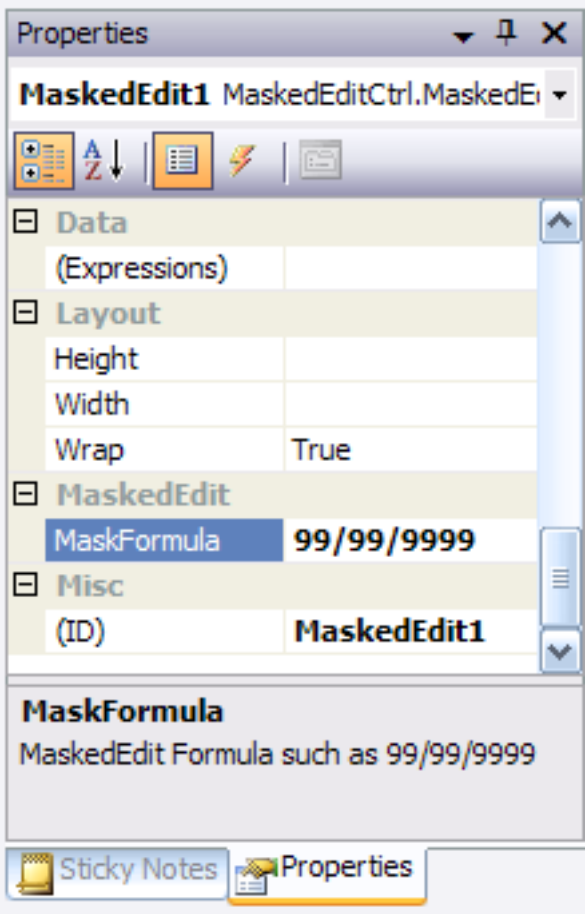
نکته جاوا اسکریپتی: علت استفاده از this.ClientID جهت معرفی نام کنترل جاری این است که هنگامیکه کنترل توسط یک master page رندر شود، ID آن توسط موتور ASP.Net کمی تغییر خواهد کرد. برای مثال myTextBox به ctl00_ContentPlaceHolder1_myTextBox تبدیل خواهد شد و اگر صرفا this.ID ذکر شده باشد دیگر دسترسی به آن توسط کدهای جاوا اسکریپت مقدور نخواهد بود. بنابراین از ClientID جهت دریافت ID نهایی رندر شده توسط ASP.Net کمک می‌گیریم.

در اینجا MaskFormula مقداری است که هنگام افزودن کنترل به صفحه می‌توان تعریف کرد.

```
[Description("MaskedEdit Formula such as 99/99/9999")]
       [Bindable(true), Category("MaskedEdit"), DefaultValue(0)]
       public string MaskFormula
       {
           get
           {
               if (ViewState["MaskFormula"] == null) ViewState["MaskFormula"] = "99/99/9999";
               return (string)ViewState["MaskFormula"];
           }
           set { ViewState["MaskFormula"] = value; }

       }
```

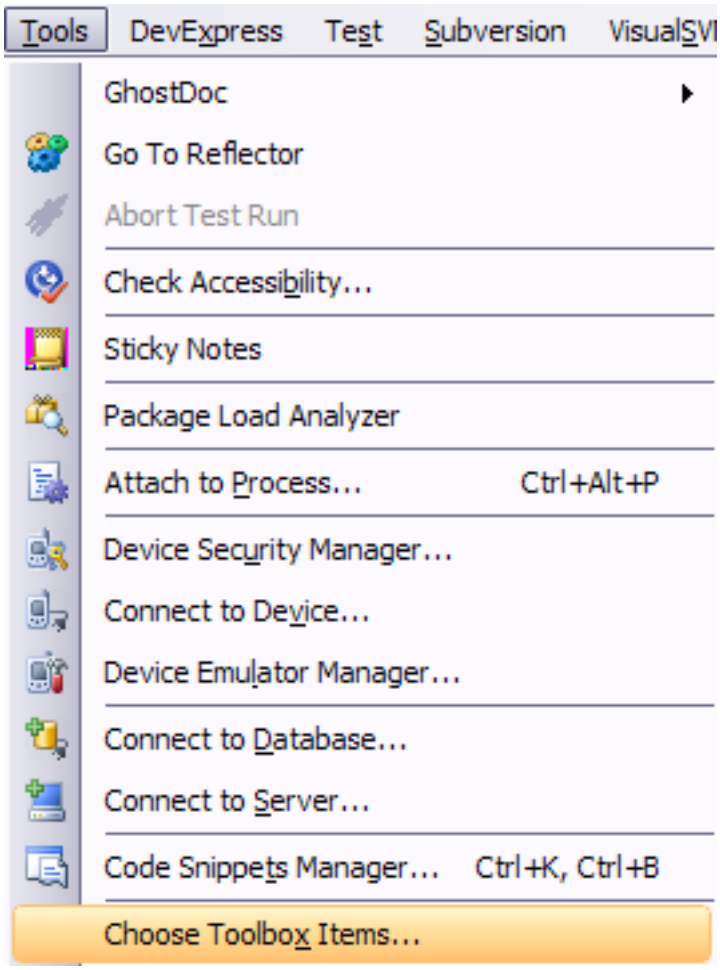
این خاصیت public هنگام نمایش در Visual studio به شکل زیر درخواهد آمد:



نکته مهم: در اینجا حتما باید از view state جهت نگهداری مقدار این خاصیت استفاده کرد تا در حین post back ها مقادیر

انتساب داده شده حفظ شوند.

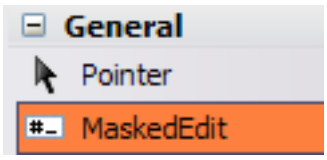
اکنون پروژه را کامپایل کنید. برای افزودن کنترل ایجاد شده به toolbox می‌توان مطابق تصویر عمل کرد:



نکته: برای افزودن آیکون به کنترل (جهت نمایش در نوار ابزار) باید: الف) تصویر مورد نظر از نوع bmp باشد با اندازه 16 در16 pixel . ب) باید آنرا به پروژه افزود و build action آن را به Embedded Resource تغییر داد. سپس آنرا در فایل AssemblyInfo.cs پروژه نیز تعریف کرد (به صورت زیر).

```
[assembly: System.Web.UI.WebResource("MaskedEditCtrl.MaskedEdit.bmp", "img/bmp")]
```

کنترل ما پس از افزوده شدن، شکل زیر را خواهد داشت:



جهت دریافت سورس کامل و فایل باینری این کنترل، اینجا کلیک کنید.

نظرات خوانندگان

نویسنده: شاهین کایست
تاریخ: ۱۳۸۹/۱۰/۲۳ ۲۲:۳۹:۳۰

سلام.
از کنترلی که طراحی کردید درون یک JQuery UI دیالوگ استفاده کردم (درون محتویات Dialog در یک Update Panel هست).
اما پس از قرار دادن کنترل Dialog از کار افتاد.
نکته : Dialog را از سمت Server پس از Postback اجرا کردم.
به نظرتون مشکل از کجا هست؟
ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۰/۲۳ ۲۲:۵۷:۴۱

سلام،
چند مورد هست:
- یکی اینکه بهتر است نسخه‌ی جدید jQuery را به این سورس اضافه و کامپایل کنید.
- مورد دیگر آشنایی با jQuery Live است : (+) ، که پس از postback ، نیاز به تزریق یا بایند مجدد یک سری اطلاعات می‌باشد و همچنین: (+)