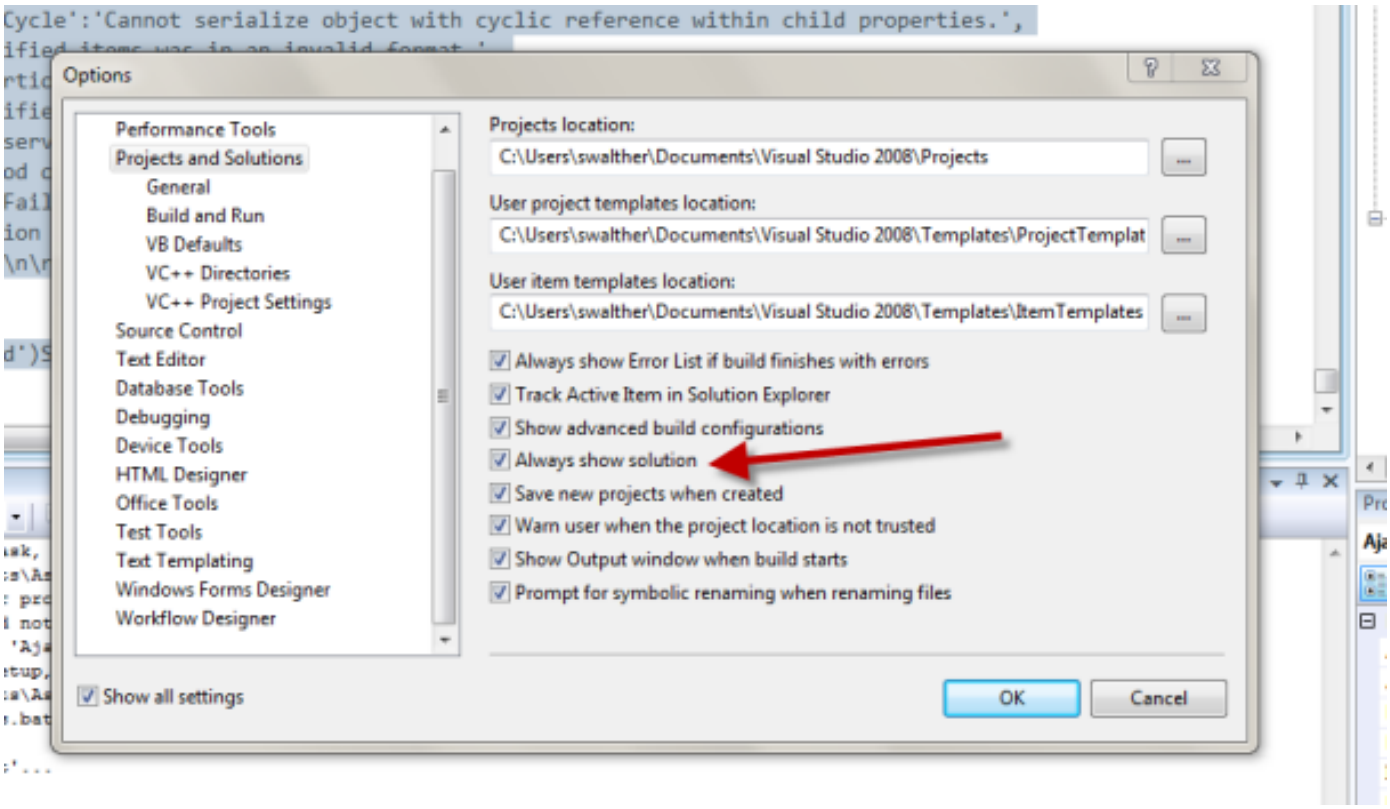
عنوان: **فشرده سازی فایل های CSS و JavaScript بصورت خودکار توسط MS Ajax Minifier**
نویسنده: علیرضا اسم‌رام
تاریخ: ۱۳۹۱/۰۴/۲۹ ۱۶:۲۹
آدرس: www.dotnettips.info
برچسب‌ها: JavaScript, Ajax, MS Ajax Minifier, CSS

با توجه به افزایش کاربرد jQuery و دیگر کتابخانه‌های جاوا اسکریپت در برنامه‌های تحت وب، یکی از چالش‌های همیشگی برنامه نویسان، فشرده سازی فایل‌های دربرگیرنده کدهای جاوا اسکریپت و شیوه نامه‌ها می باشد. برای این منظور راه‌های مختلفی مانند استفاده از ابزارهای آنلاین مانند این + و این + وجود دارند. اما یک روش خودکار هم وجود دارد که در زمان Build پروژه‌های دات نت می‌توان از آن بهره گرفت.

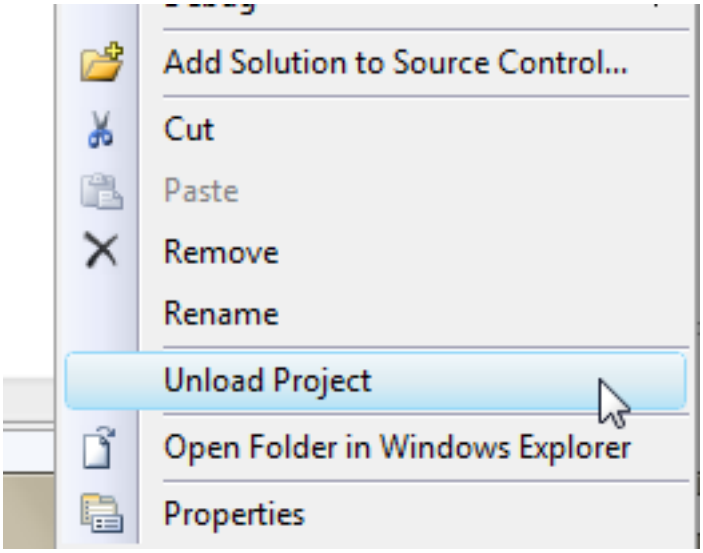
Microsoft Ajax Minifier

یک ابزار رایگان جهت فشرده سازی فایل‌های جاوا اسکریپت و شیوه نامه‌ها است. شما می‌توانید این ابزار را از صفحه خانگی آن در سایت asp.net دریافت کنید. جهت استفاده از این ابزار می‌توان از طریق خط فرمان عمل کرد. اما روش ساده‌تر که هدف اصلی این مطلب است به شرح زیر است:

1. در VisualStudio.NET از طریق منو به مسیر **Tools, Options, Projects and Solutions** بروید و گزینه **Always show solution** را تیک بزنید.



2. از Solution Explorer بر روی عنوان پروژه کلیک راست کرده و گزینه **Unload Project** را انتخاب نمایید.



3. مجدداً روی عنوان پروژه کلیک راست کرده و گزینه **Edit** را انتخاب کنید و دستورات زیر را قبل از بسته شدن تگ Project اضافه کنید:

```
<Import Project="$(MSBuildExtensionsPath)\Microsoft\MicrosoftAjax\ajaxmin.tasks" />
<Target Name="AfterBuild">
<ItemGroup>
  <JS Include="**\*.js" Exclude="**\*.min.js;Scripts\*.js" />
</ItemGroup>
<ItemGroup>
  <CSS Include="**\*.css" Exclude="**\*.min.css" />
</ItemGroup>
<AjaxMin
    JsSourceFiles="@(JS)"  JsSourceExtensionPattern="\.js$" JsTargetExtension=".min.js"
    CssSourceFiles="@(CSS)" CssSourceExtensionPattern="\.css$" CssTargetExtension=".min.css"  />
</Target>
```

4. دوباره بر روی عنوان پروژه کلیک راست کرده و گزینه **Reload Project** را انتخاب کنید.

توجه کنید که با این کار ما یک MSBuild task با عنوان ajaxmini به پروژه اضافه کردیم. این وظیفه که در زمان Build پروژه اجرا خواهد شد فایل‌های جاوا اسکریپت را فشرده و با پسوند .min.js و همچنین فایل‌های CSS را پس از فشرده سازی با پسوند .min.css در همان مسیر فایل مادر بطور خودکار ذخیره می‌کند.

نکته:

اگر به دستورات تنظیمات فوق نگاه دقیقتری بیندازیم، متوجه عبارات **Include** و **Exclude** می شویم. توسط این دو صفت شما می‌توانید الگوهایی را جهت فشرده سازی و یا عدم فشرده سازی تعریف کنید. بدین معنا که توسط الگوی‌های ذکر شده در تنظیمات فوق از فشرده سازی فایل‌های با پسوند .min.css و .min.js خودداری می‌شود.

در این شرایط در حین توسعه برنامه، شما می‌توانید از فایل‌های با کد خوانا استفاده نمایید و زمان انتشار و Build پروژه بصورت خودکار آنها را با فایل‌های فشرده جایگزین کنید.

این ابزار تمامی فضاهای خالی، ';' و '{ }'های اضافی و توضیحات را از کدهای شما حذف می‌کند. متغیرها و توابع شما را به اسامی کوچک‌تر تغییر نام می‌دهد. و ...

همچنین شما از کتابخانه این پروژه می‌توانید در زمان اجرا و سورس برنامه خود استفاده کنید. جهت اطلاعات بیشتر می‌توانید به سایت مربوطه مراجعه نمایید.

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۲۹ ۲۲:۴۲

ممنون. من از این کتابخانه استفاده می‌کنم:

Yahoo! UI Library: YUI Compressor for .Net

```
using System.Globalization;
using System.IO;
using System.Text;
using Yahoo.Yui.Compressor;

namespace Deploy.Core
{
    public static class CompressCssJs
    {
        public static void Compress(string file)
        {
            var ext = Path.GetExtension(file).ToLower();
            switch (ext)
            {
                case ".css":
                    compressCss(file);
                    break;
                case ".js":
                    if (!file.ToLower().EndsWith(".min.js") && !file.ToLower().EndsWith(".pack.js"))
                        compressJs(file);
                    break;
            }
        }

        static void compressCss(string file)
        {
            var css = File.ReadAllText(file);
            var compressedCss = new CssCompressor().Compress(css);
            File.WriteAllText(file, compressedCss, Encoding.UTF8);
        }

        static void compressJs(string file)
        {
            var js = File.ReadAllText(file);

            var compressedJavaScript = new JavaScriptCompressor
            {
                CompressionType = CompressionType.Standard,
                DisableOptimizations = false,
                Encoding = Encoding.UTF8,
                LineBreakPosition = -1,
                ObfuscateJavascript = true,
                PreserveAllSemicolons = false,
                ThreadCulture = CultureInfo.CurrentUICulture,
                IgnoreEval = false,
                LoggingType = LoggingType.None
            }.Compress(js);
            File.WriteAllText(file, compressedJavaScript, Encoding.UTF8);
        }
    }
}
```

نحوه استفاده از اون رو باکدنویسی در بالا ملاحظه می‌کنید (ملاحظات utf8 و زبان فارسی هم در آن لحاظ شده).
کاری که هنگام ارائه نهایی انجام می‌دم، اسکن فایل‌های نهایی و بررسی پسوندها و سپس استفاده از متد Compress فوق روی فایل‌های اسکریپت و css یافت شده است.

نویسنده: علیرضا اسم‌رام
تاریخ: ۱۳۹۱/۰۴/۳۰ ۹:۶

ضمن تشکر از آقای نصیری بخاطر معرفی روش دیگر، توجه فرمایید من تنها یک معرفی اجمالی درباره این ابزار داشتم. این ابزار

امکانات دیگری نیز دارد که بطور خاص برخی از آنها به شرح زیر است:
آنالیز فایل‌های جاوا اسکریپت و اعلام هشدارهای مناسب به برنامه نویس. +
اگر از فایل‌های RESX جهت چند زبانه کردن سایت استفاده می‌کنید این ابزار بسیار کارآمد خواهد بود. +
همچنین این ابزار نیز از UTF-8 بخوبی پشتیبانی می‌کند.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۳۰ ۹:۱۸

ممنون. ابزار یاهو هم امکان آنالیز (LoggingType) را دارد. فقط در اینجا چون مد نظر من نبوده به none تنظیم شده.

نویسنده: علیرضا اسم‌رام
تاریخ: ۱۳۹۱/۰۴/۳۰ ۹:۳۰

به نظر میاد نسبت به هم در یک سطح باشند. سپاس از مطالب به موقع و بجای شما.

نویسنده: رضا.ب
تاریخ: ۱۳۹۱/۰۴/۳۰ ۱۹:۶

فکر میکنم از این رهیافت در سلوشن‌های مرجح مایکروسافت نشه یاد کرد. در asp.net mvc 4 دو تکنیک Bundle کردن (دسته کردن فایل‌های مشابه در یک دسته) و Minify کردن (از بین بردن فواصل اضافه و کوچک کردن نام متغییرها و حذف کامنت‌ها) اضافه شده. که این کار رو به صورت توکار خود فریم‌ورک انجام میده فقط لازمه با دستورات Syte.Render و Script.Render اونارو آدرس دهی کنیم.
همچنین این قابلیت وجود دارد که در زمان debug فایل‌ها رو خانا و قابل‌فهم مشاهده کنید. در یکی از سری پست‌های آقای شهروز جعفری ^ این مهم اشاره کردن. هرچند پست ایشان اندکی قدیمی است(موقع نگارش beta مطلب منتشر شد که حالا با آمدن نگارش RC اندکی فرق کرده است)

نویسنده: علیرضا اسم‌رام
تاریخ: ۱۳۹۱/۰۴/۳۰ ۱۹:۳۵

با تشکر از آقای رضا.ب....
مطلب آقای جعفری را مطالعه کردم. آیا امکان استفاده از فایل‌های Resource جهت Localization در این تکنیک وجود دارد؟
همچنین آیا فایل‌ها اسکریپت آنالیز می‌شوند و ما از هشدارهای مناسب مطلع می‌شویم؟

نویسنده: ایمان اسلامی
تاریخ: ۱۳۹۱/۰۴/۳۰ ۲۰:۳۹

با تشکر از دوستان بابت تمامی روشهای بحث شده.
میخواستم بدونم برای فشرده سازی یک web site قبل از upload با gzip هم میشه اطلاعات مختصری بدهید.
با تشکر

نویسنده: علیرضا اسم‌رام
تاریخ: ۱۳۹۱/۰۴/۳۰ ۲۳:۳

تا جایی که من مطلع هستم از طریق تنظیمات IIS می‌توان پاسخ به درخواست‌ها را (شامل فایل‌های استاتیک و داینامیک) به کمک gzip فشرده کرد. اما اگر اطلاعاتم صحیح باشد این موضوع کمی بار CPU را افزایش می‌دهد، هر چند گاهی تا 75% حجم اطلاعات رد و بدل شده را کاهش می‌دهد.
برای اطلاعات بیشتر در مورد تنظیمات IIS6 می‌توانید به + و + مراجعه کنید.
همچنین اگر روی IIS7 به بعد میزبانی می‌شوید می‌توانید درون فایل Web.Config و درون تگ system.webServer تنظیمات زیر را اضافه کنید:

```
<httpCompression directory="%SystemDrive%\inetpub\temp\IIS Temporary Compressed Files">
    <scheme name="gzip" dll="%Windir%\system32\inetsrv\gzip.dll" staticCompressionLevel="9"
dynamicCompressionLevel="4" />
    <scheme name="deflate" dll="%Windir%\system32\inetsrv\gzip.dll" staticCompressionLevel="9"
dynamicCompressionLevel="4" />
    <dynamicTypes>
        <add mimeType="text/*" enabled="true" />
        <add mimeType="message/*" enabled="true" />
        <add mimeType="application/x-javascript" enabled="true" />
        <add mimeType="application/atom+xml; charset=utf-8" enabled="true" />
        <add mimeType="*/*" enabled="false" />
    </dynamicTypes>
    <staticTypes>
        <add mimeType="text/*" enabled="true" />
        <add mimeType="message/*" enabled="true" />
        <add mimeType="application/javascript" enabled="true" />
        <add mimeType="*/*" enabled="false" />
    </staticTypes>
</httpCompression>
```

(برای مطالعه بیشتر این + را ببینید.)

نویسنده: ایمان اسلامی
تاریخ: ۱۳۹۱/۰۴/۳۱ ۸:۱۴

ممنون از جواب خوبتون
پس بطور کلی بهتر است که اینکارو انجام ندهیم؟

نویسنده: علیرضا اسم‌رام
تاریخ: ۱۳۹۱/۰۴/۳۱ ۹:۵۱

من پیشنهاد می‌کنم اگر شرایطش رو دارید، امتحانش کنید. حتی برای تجربه شده! و خوشحال میشم نتیجه تجربه تان را با ما به اشتراک بگذارید.

نویسنده: رضا.ب
تاریخ: ۱۳۹۱/۰۴/۳۱ ۱۹:۱

تا جایی که اطلاع دارم، تکنیک B/M فقط باعث کاهش میزان درخواست‌های HTTP بر روی وب سرور میشود. (آموزش نسخه MVC 4 RC ^)

نویسنده: صابر فتح اللهی
تاریخ: ۱۳۹۱/۰۵/۱۶ ۱۱:۱۹

سلام
مهندس نصیری این روشی که شما فرمودین
در زمان اجرا خروجی فشرده میکنه یا اینکه نه فشرده میکنه و همونجا هم ذخیره میکنه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۶ ۱۳:۵۴

هنگام ارائه نهایی (در پوشه‌ای جداگانه از فایل‌های اصلی) یکبار روی فایل‌ها اجرا می‌شود.

نویسنده: صابر فتح اللهی
تاریخ: ۱۳۹۱/۰۹/۰۶ ۸:۳۹

سلام

به نظر شما بهتر نیست این کدها در یک هندلر استفاده بشه که در زمان اجرا خروجی فشرده شده برای کاربر ارسال بشه با توجه به کش شدن این نوع فایل‌ها ایا سربار سیستم اضافه میشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۰۶ ۹:۵۶

سیستم bundling جدید MVC4 در زمان اجرا این‌کارها را انجام می‌دهد. کش شدن هم سربار سیستم رو کم می‌کنه چون درخواست جدیدی به سرور ارسال نخواهد شد.

نویسنده: سیاه
تاریخ: ۱۳۹۱/۱۱/۱۸ ۵:۲۷

سلام؛ راهی وجود داره که لینک فایلهای css و js رو در source page مرورگر غیرفعال کرد ؟
و امکان اینکه فرد با کلیک روی لینک محتوای فایلهای css و js رو نبینه .
با تشکر .

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۱۸ ۹:۹

خیر. مرورگر باید به این لینک‌ها دسترسی داشته باشه برای رندر سایت. به همین نحو این دسترسی برای تمام دنیا وجود دارد.

نویسنده: مرتضی مختاری
تاریخ: ۱۳۹۲/۰۳/۲۳ ۲۳:۵۲

سلام؛ اگه از کش کردن اطلاعات استاتیک استفاده کنیم دیگه نیازی به minify کردن فایل‌ها داریم یا خیر؟