**[优化网站设计（四）：对资源启用压缩](http://www.cnblogs.com/chenxizhang/archive/2013/05/01/3052881.html)**

**前言**

网站设计的优化是一个很大的话题,有一些通用的原则,也有针对不同开发平台的一些建议。这方面的研究一直没有停止过，我在不同的场合也分享过这样的话题。

作为通用的原则，雅虎的工程师团队曾经给出过35个最佳实践。这个列表请参考

**Best Practices for Speeding Up Your Web Site** [**http://developer.yahoo.com/performance/rules.html**](http://developer.yahoo.com/performance/rules.html)

同时，他们还发布了一个相应的测试工具Yslow <http://developer.yahoo.com/yslow/>

我强烈推荐所有的网站开发人员都应该学习这些最佳实践，并结合自己的实际项目情况进行应用。

接下来的一段时间，我将结合ASP.NET这个开发平台，针对这些原则，通过一个系列文章的形式，做些讲解和演绎，以帮助大家更好地理解这些原则，并且更好地使用他们。

**准备工作**

为了跟随我进行后续的学习，你需要准备如下的开发环境和工具

1. Google Chrome 或者firefox ，并且安装 Yslow这个扩展组件.请注意，这个组件是雅虎提供的，但目前没有针对IE的版本。
   1. <https://chrome.google.com/webstore/detail/yslow/ninejjcohidippngpapiilnmkgllmakh>

Technorati Tags: [Performance](http://technorati.com/tags/Performance),[Web design](http://technorati.com/tags/Web+design),[ASP.NET](http://technorati.com/tags/ASP.NET)

* 1. <https://addons.mozilla.org/en-US/firefox/addon/yslow/>
  2. 你应该对这些浏览器的开发人员工具有所了解，你可以通过按下F12键调出这个工具。

1. Visaul Studio 2010 SP1 或更高版本，推荐使用Visual Studio 2012
   1. <http://www.microsoft.com/visualstudio/eng/downloads>
2. 你需要对ASP.NET的开发基本流程和核心技术有相当的了解，本系列文章很难对基础知识做普及。

**本文要讨论的话题**

这一篇，我们要讨论的是对资源启用压缩的话题。我们知道，不光我们如何[减少请求数](http://www.cnblogs.com/chenxizhang/archive/2013/04/29/3050839.html)，或者[使用CDN](http://www.cnblogs.com/chenxizhang/archive/2013/04/30/3051686.html)，以及[使用缓存](http://www.cnblogs.com/chenxizhang/archive/2013/04/30/3052440.html)，有一个事实是我们无法避免的：内容总是需要从服务器传输到客户端，那怕次数是少一些。那么，如果希望这个传输的过程尽量地快，我们应该会很自然地想到，能否将传输的内容体积减小呢？

要回答这样的一个问题，我们通常就会使用到压缩技术。关于这一条原则的理论和概念，请参考 <http://developer.yahoo.com/performance/rules.html#gzip>

压缩并不那么简单，其实它包含了一对操作：压缩和解压缩。换句话说，在我们今天讨论的网站优化中使用压缩技术，不仅仅需要考虑服务器端对内容进行压缩，还要考虑客户端（浏览器）对内容进行解压缩。反过来也是如此。所以，这里就会有一个问题，我们必须要使用大多数浏览器都能接受的压缩算法。由于浏览器的多样性，通常是由浏览器在发起请求的时候，显式地表明自己接受那些压缩算法，然后服务器检查这些设置，再确认自己是否能利用这些算法进行压缩（或者解压缩），如果不能，则宁愿不进行压缩，直接返回原始的内容。

所以，在[HTTP 1.1](http://www.w3.org/Protocols/rfc2616/rfc2616.html)中规定，浏览器在发起请求的时候，可以通过下面这个Request Header来表明自己支持的压缩算法（可以有多个）

Accept-Encoding: gzip, deflate

然后，服务器在发送响应的时候，也可以通过下面这个Response Header来表明此次响应是否使用了某种算法（肯定只有一个）

Content-Encoding: gzip

作为程序员，我们知道其实还有很多其他的算法，但是确实从通用层面考虑，使用最多的是[Gzip](http://www.ietf.org/rfc/rfc1952.txt)。 其他能与其相提并论的还有deflate，但还是Gzip用的最多。

**哪些资源适合做压缩**

1. 静态网页（HTML,HTM）
2. 文本文件（TEXT,XML等）
3. 脚本文件（JAVASCRIPT)
4. 样式文件（CSS)

**哪些资源不适合做压缩**

1. 图片（JPG,GIF,PNG)
2. 特殊组件（FLASH, XAP）

**如何做压缩**

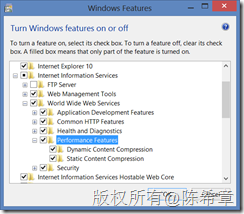
要实现压缩功能并不难，现代的一些Web 服务器都内置支持这个特性。针对微软的IIS 7.0或者更高版本的话，可以通过参考下面的文章进行配置

**Configuring HTTP Compression in IIS 7**

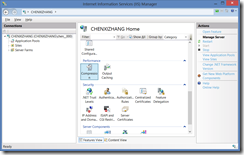
<http://technet.microsoft.com/en-us/library/cc771003(v=WS.10).aspx>

我来对其做一些总结和演示

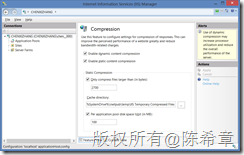
IIS 7.0内置支持Gzip压缩，这个可以通过在安装IIS的时候进行选择

[](http://images.cnitblog.com/blog/9072/201305/01112658-2a8fd419888c4b3b8339aa1744305b7a.png)

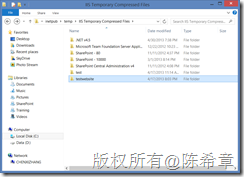
正确安装之后，在管理工具中，就可以看到这样一项功能

[](http://images.cnitblog.com/blog/9072/201305/01112703-ca9cfd706ca548319b2b52d30a3c793e.png)

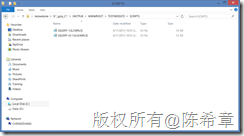
IIS 7.0 支持两种方式的压缩：静态压缩和动态压缩。所谓静态压缩，就是对相对较大的内容，IIS 7.0会将它们压缩成一个新文件，并且缓存在磁盘上（可以通过下面这个界面配置多大的文件要进行静态压缩，并且放在哪个目录中），而动态压缩就是对于某些提交小的内容，直接在运行时进行动态压缩，不在磁盘上进行缓存（这种做法会带来CPU的一些额外的负担）

[](http://images.cnitblog.com/blog/9072/201305/01112708-9dc0c0e9ea82476f94e62d897a150331.png)

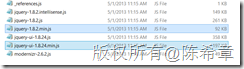
这个缓存的目录中会自动为每个应用程序池(Application Pool)创建一个子目录，以便保存那些静态压缩的文件

[](http://images.cnitblog.com/blog/9072/201305/01112709-f4aa15504ac9481a9e2ed2fc7f67287f.png)

那么这种压缩到底能有多少收益呢？我们可以看看下面这个截图

[](http://images.cnitblog.com/blog/9072/201305/01112711-a004ce68657d4fe0a9cbdbbea7f966a1.png)

上图中看到，在我的一个测试网站中，它对两个JAVASCRIPT文件做了压缩，压缩之后的尺寸分别为33KB和51KB，而这两个文件的原始文件大小其实是92KB和197KB，我们可以由此看出来，压缩比高达65%和74%。这是相当可观的一个收益，而你要做的仅仅是启用压缩即可。

[](http://images.cnitblog.com/blog/9072/201305/01112712-4556a509e8264034a1002b2b50f5c2e7.png)

实际上这个文件已经被压缩成下面这样了（不再是纯文本的脚本了）

[](http://images.cnitblog.com/blog/9072/201305/01112715-ae2c9210c02c43c69ce57d12de74d223.png)

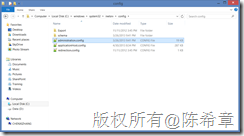
不可否认的是，压缩和解压缩肯定会对CPU带来一些额外的负担的，但通常情况下，这个代价是很小的，尤其是与收益比较起来的话。

等等！我们好像漏了一个很重要的话题:上面的界面中，我们知道如何启用压缩，但却没有看到

1. 如何设置哪些文件应该如何压缩（无论静态还是动态）。
2. 到底是采用什么算法压缩

这个细节被隐藏在下面的配置文件中

C:\windows\system32\inetsrv\config\applicationHost.config

[](http://images.cnitblog.com/blog/9072/201305/01112718-26949ed0942e4710b72fe3a7bcf8dcf5.png)

打开这个文件，搜索一下httpCompression，你可以找到如下的配置信息，当然，你可以在这个基础上做一些修改，前提是你先看懂它们。实际上不难，对吧

[](http://images.cnitblog.com/blog/9072/201305/01112724-1807401cec1f49c1ae8c95b99722e2a4.png)

Technorati Tags: [性能优化](http://technorati.com/tags/%e6%80%a7%e8%83%bd%e4%bc%98%e5%8c%96),[performance](http://technorati.com/tags/performance),[ASP.NET](http://technorati.com/tags/ASP.NET),[web design](http://technorati.com/tags/web+design)