**ashx和aspx的具体功能相同，而ashx更简单，不会对web页面的控件进行解析，因此响应快，消耗资源小。**

**ashx是什么文件?**

.ashx 文件用于写web handler的。.ashx文件与.aspx文件类似，可以通过它来调用HttpHandler类，它免去了普通.aspx页面的控件解析以及页面处理的过程。其实就是带HTML和C#的混合文件。

.ashx文件适合产生供浏览器处理的、不需要回发处理的数据格式，例如用于生成动态图片、动态文本等内容。

ashx文件是.net 2.0新加的文件类型(其实在.net 1.0下已经可用,但是没有公开提供).

ashx文件和aspx文件有什么不同? 我们先新建一个ashx文件看看:

**代码示例：**当然你完全可以用.aspx 的文件后缀。使用.ashx 可以让你专注于编程而不用管相关的WEB技术。.ashx必须包含IsReusable. 如下例所示

**[csharp]** [view plain](http://blog.csdn.net/whatday/article/details/54893889) [copy](http://blog.csdn.net/whatday/article/details/54893889)

1. <% @ webhandler language="C#" **class**="AverageHandler" %>
2. **using** System;
3. **using** System.Web;
4. **public** **class** AverageHandler : IHttpHandler
5. {
6. **public** **bool** IsReusable
7. {
8. **get** { **return** **true**; }
9. }
10. **public** **void** ProcessRequest(HttpContext ctx)
11. {
12. ctx.Response.Write("hello");
13. }
14. }

.ashx比.aspx的好处在于不用多一个html

看,比aspx简洁多了吧.只有一个文件,没有后台cs文件(基于代码安全考虑,后边我们会自己添加这个文件).ashx对比aspx文件,就好像 少了cs文件.其实这就是ashx和aspx不同的地方,因为aspx要将前后台显示和处理逻辑分开,所以就弄成了两个文件,其实,在最终编译的时 候,aspx和cs还是会编译到同一个类中去.这中间就要设计html的一些逻辑处理.而ashx不同,它只是简单的对web http请求的直接返回你想要返回的结果.比aspx少处理了html的过程.理论上,比aspx要快.

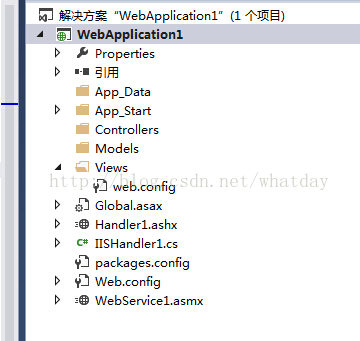
看看.net config文件中对两个文件类型请求的配置吧

<add path="\*.aspx" verb="\*" type="System.Web.UI.PageHandlerFactory" validate="True" />  
<add path="\*.ashx" verb="\*" type="System.Web.UI.SimpleHandlerFactory" validate="True" />

可以看到两个文件处理的类不一样(ashx处理的类叫SimpleHandleFactory,既然叫Simple,应该处理过程也比较 Simple.响应速度也应该快点吧:)

鼓捣完了你就大概明白了ashx不过.net里面自定义好的一个请求格式,ashx文件类的文件的处理过程要比aspx要简单得多.

=========================================================

.NET里面webform的后缀是aspx

WCF和WebService的后缀是asmx

然后今天拿到一个客户端代码，调用服务端，服务端后缀是ashx瞬间傻蛋了，.NET我不知道的组件真多。

四个疑问：

1、什么时候用

2、优缺点

3、简单实现机制

4、简单DEMO

**1、什么时候用**

    虽然通过标准的方式可以创建处理程序，但是实现的步骤比较复杂，为了方便网站开发中对处理程序的应用，从Asp[**.NET**](http://lib.csdn.net/base/dotnet) 2.0开始，asp[**.Net**](http://lib.csdn.net/base/dotnet)提供了称为一般处理程序的处理程序，允许我们使用比较简单的方式定义扩展名为ashx的专用处理程序。

    对于asp[**.NET**](http://lib.csdn.net/base/dotnet)网站来说，网站最觉的处理结果就是HTML网页，生成网页的工作通常使用扩展名为aspx的Web窗体来完成。对于处理结果不是HTML的请求，都可以通过一般处理程序完成。例如生成RSS Feed、XML、图片等。  
    一般处理程序是asp[**.Net**](http://lib.csdn.net/base/dotnet)网站中最为简单、高效的处理程序，在处理返回类型不是HTML的请求中有着重要的作用。

    不知道我的理解正不正确，那感觉asmx适合作为C/S的服务器端来使用。

**2、优缺点**

     通常是实现IHttpHandler接口，因为不必继承自Page类，所以没有那么多事件需要处理，不必消耗太多资源，所以性能方面要比aspx高

**3、简单实现机制**

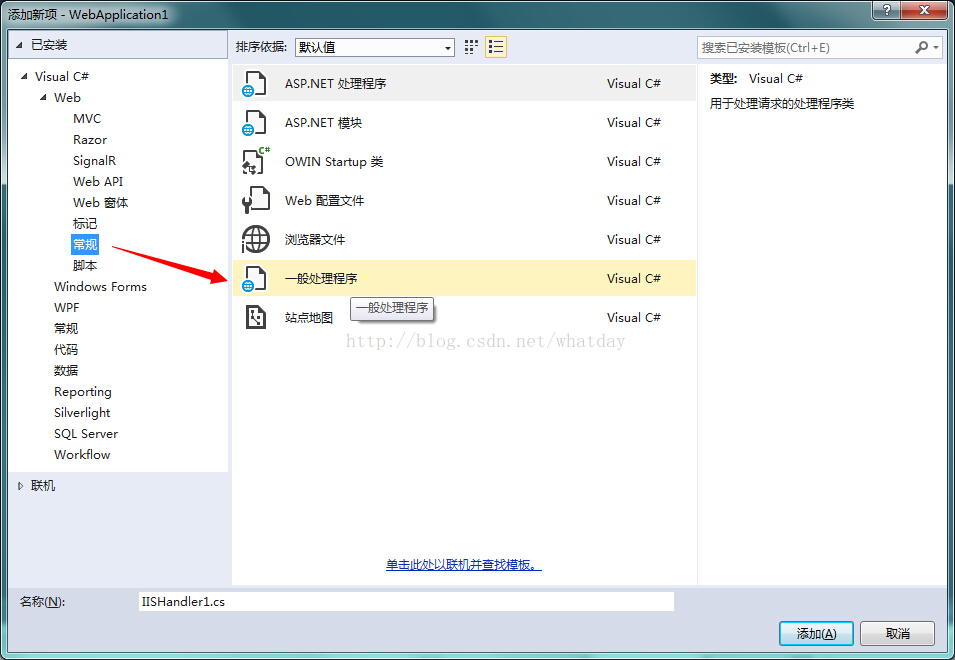
.ashx 文件用于写web handler的。.ashx文件与.aspx文件类似，可以通过它来调用HttpHandler类，它免去了普通.aspx页面的控件解析以及页面处理的过程。其实就是带HTML和C#的混合文件。

.ashx文件适合产生供浏览器处理的、不需要回发处理的数据格式，例如用于生成动态图片、动态文本等内容。

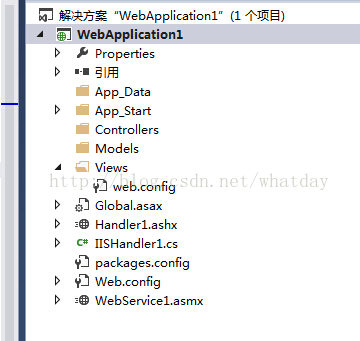
4、简单DEMO

在VS2013下，右键项目-->添加-->一般处理程序，就添加成功了，不知道是不是我添加过，所以在右键里面有。

另一种添加方式是右键项目-->添加-->新建项，然后如下图所示。



但我诧异的发现两种添加方式添加出来的文件图标不一样。



第一种方式添加图标和最下方的WebService相同，而第二种添加方式只添加了一个类。

先不管第二种，第一种生成的代码有两个文件

Handler1.ashx

**[plain]** [view plain](http://blog.csdn.net/wanda39kela/article/details/46118839) [copy](http://blog.csdn.net/wanda39kela/article/details/46118839)

1. <%@ WebHandler Language="C#" CodeBehind="Handler1.ashx.cs" Class="WebApplication1.Handler1" %>

Handler1.ashx.cs

**[csharp]** [view plain](http://blog.csdn.net/wanda39kela/article/details/46118839) [copy](http://blog.csdn.net/wanda39kela/article/details/46118839)

[在CODE上查看代码片](https://code.csdn.net/snippets/677614)

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.Linq;
4. **using** System.Web;
6. **namespace** WebApplication1
7. {
8. /// <summary>
9. /// Handler1 的摘要说明
10. /// </summary>
11. **public** **class** Handler1 : IHttpHandler
12. {
14. **public** **void** ProcessRequest(HttpContext context)
15. {
16. context.Response.ContentType = "text/plain";
17. context.Response.Write("Hello World");
18. }
20. **public** **bool** IsReusable
21. {
22. **get**
23. {
24. **return** **false**;
25. }
26. }
27. }
28. }

接下来就是怎么访问这个文件，我直接点F5运行项目，然后在浏览器地址栏加上路径，我整个路径是这样的<http://localhost:6988/Handler1.ashx>

然后成功看到浏览器上一个光秃秃的Hello World