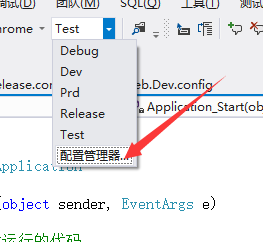
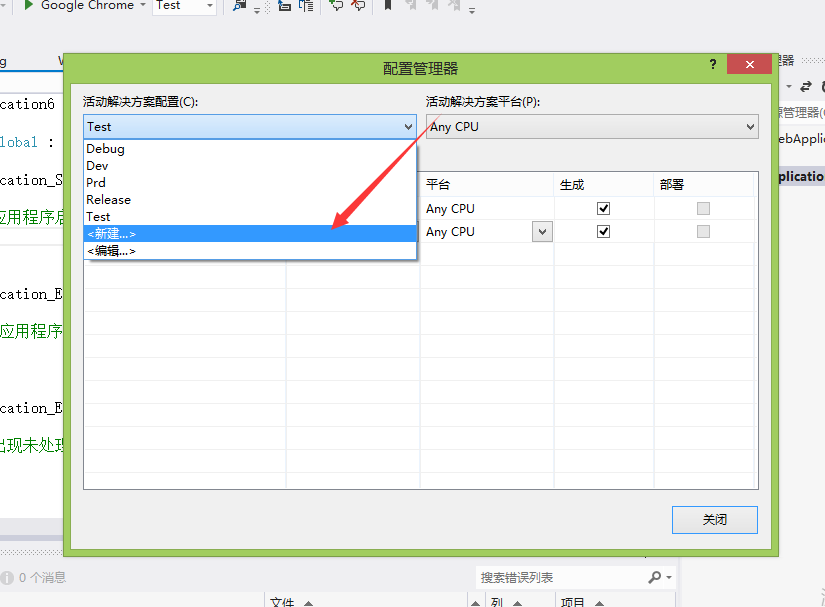
**Web.config 多环境配置(Web.config Transformations)**

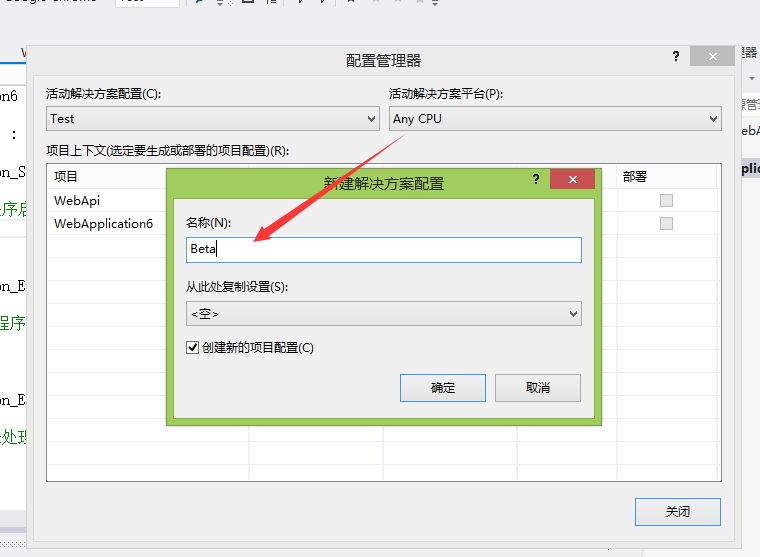
**Step1：新建配置1**



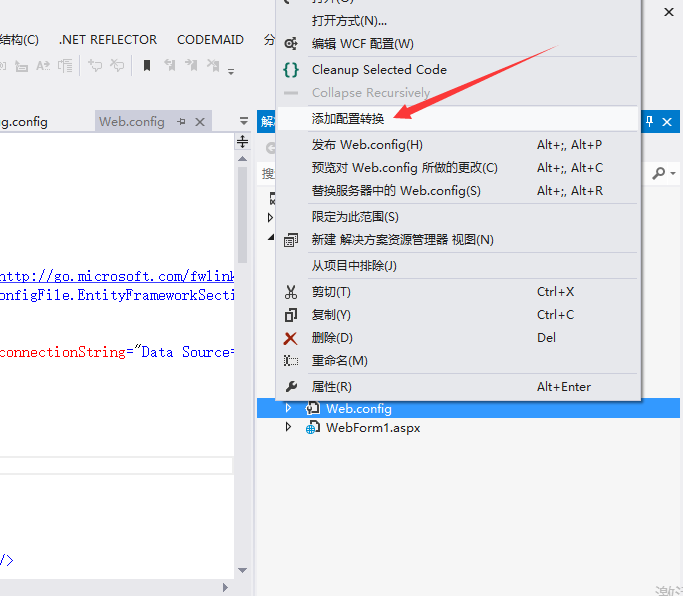
**Step2：新建配置2**

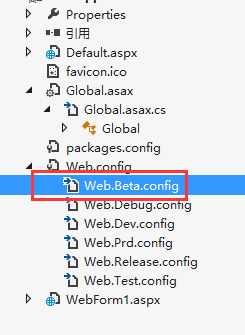


**Step3：新建配置3**



**Setp4：添加配置转换**

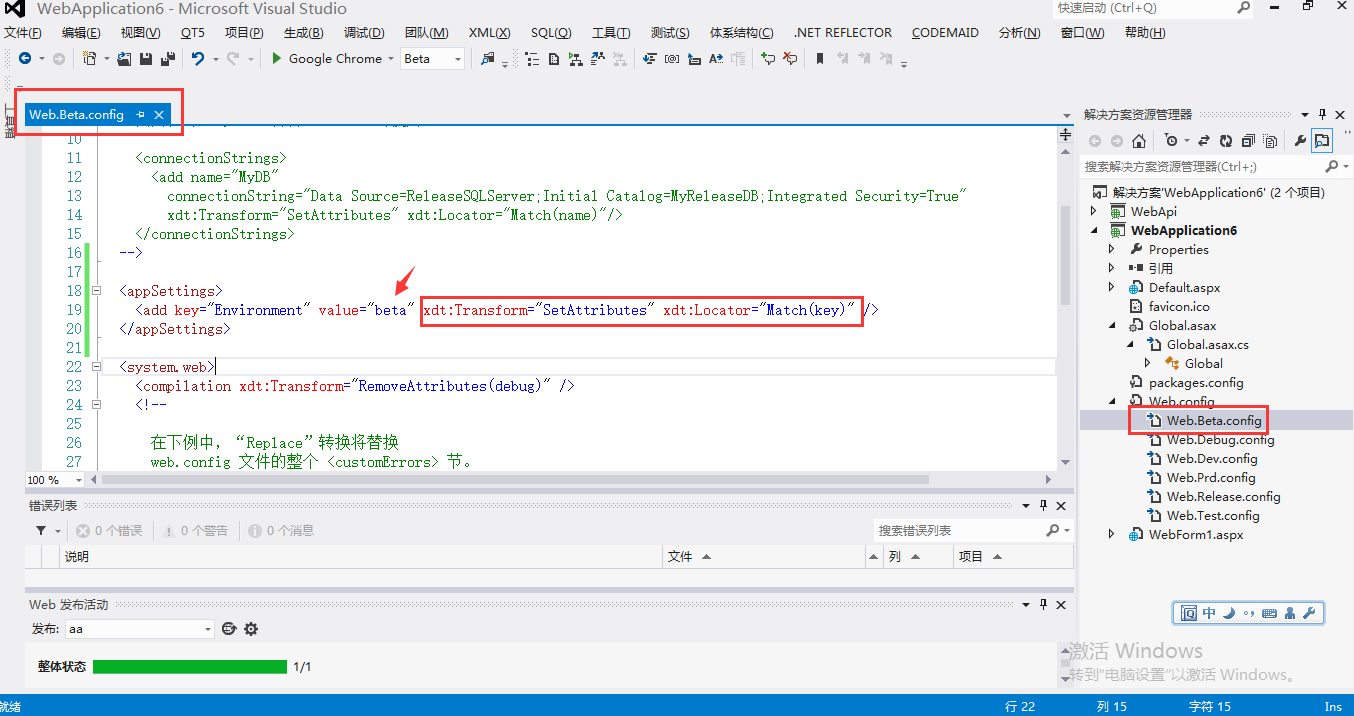




**Setp5：修改Web.Beta.config**



Web.config文件



Web.Beta.config文件

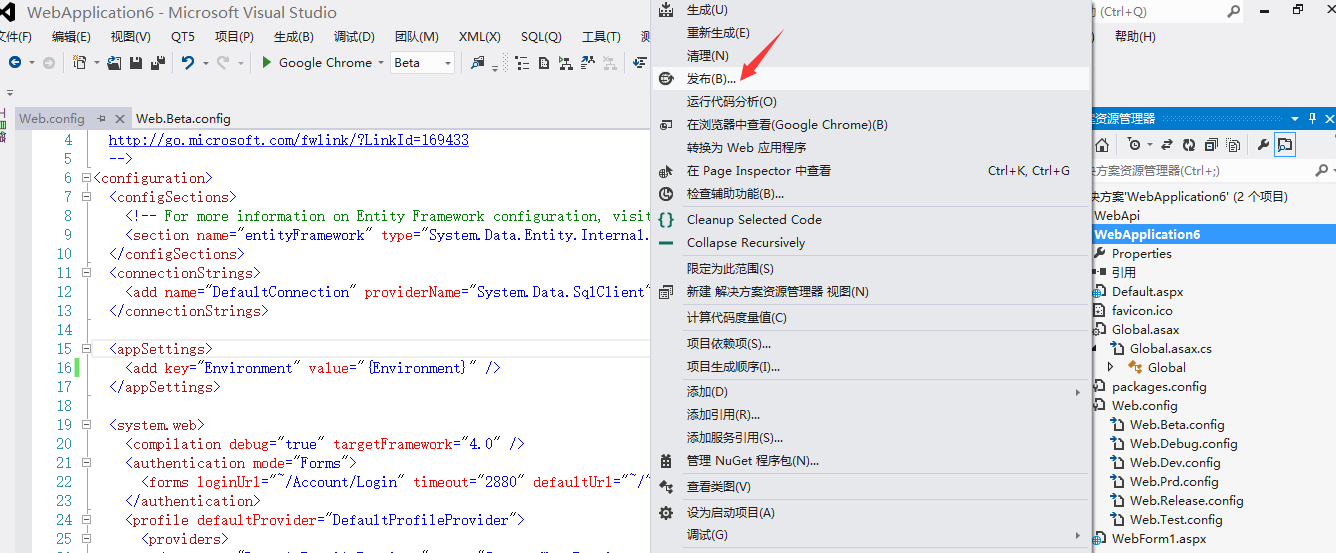
其目的是在发布时设置web.Config 文件中对应节点Environment的Value，前提是web.Config要有Environment节点，不然无效。

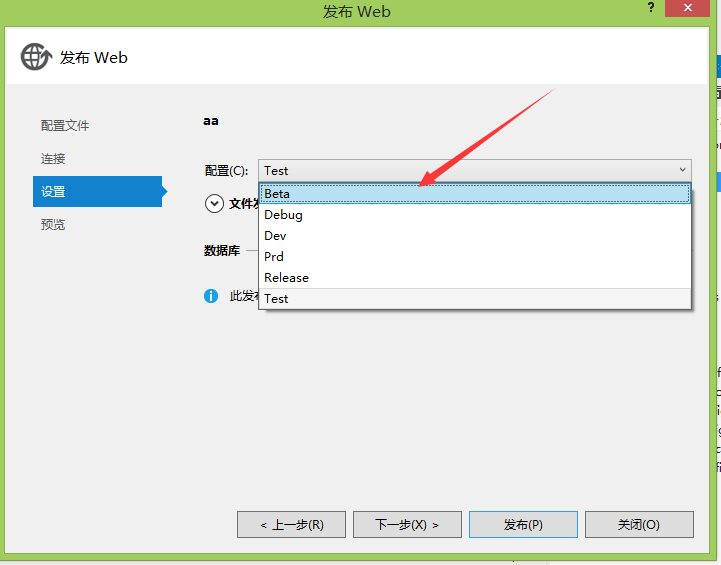
xdt:Locator 作用是定位，xdt:Transform作用是转换

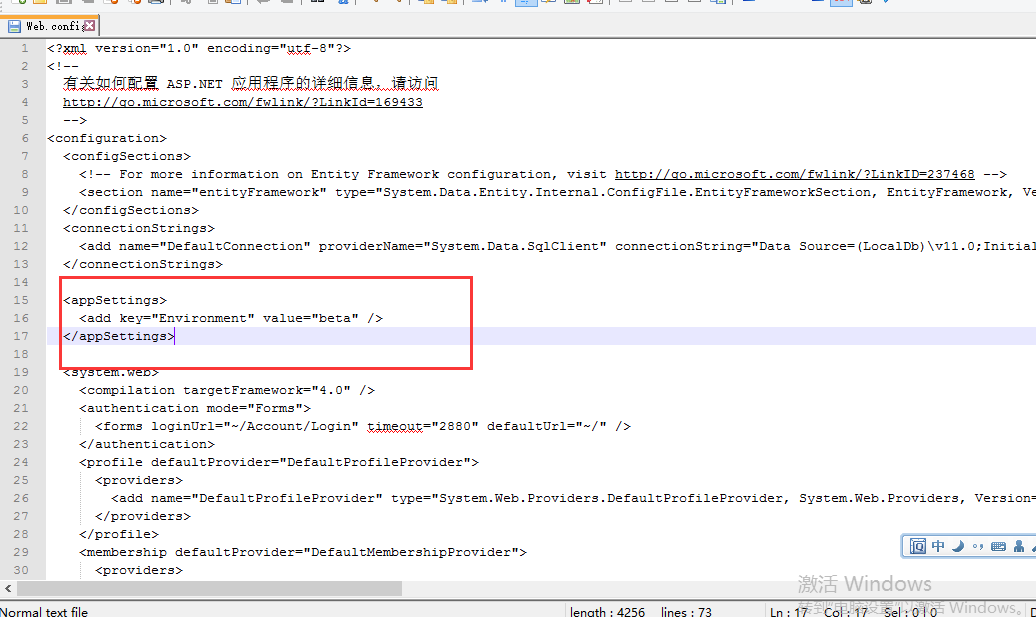
这两个xdt属性的代码意味是：Match 到一个名叫key 的Attribute。

我们也可以

**Setp6：选择相应的配置发布**



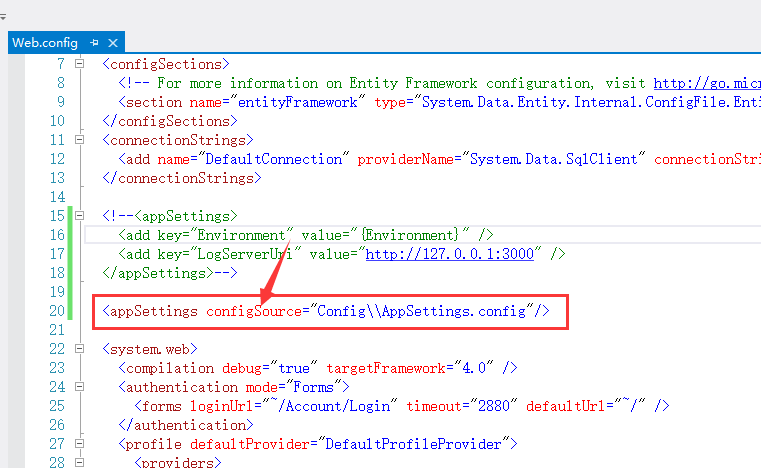




发布完成后的web.config文件

**----------------------------------我是分割线---------------------------------------**

很多时候我们的Appsettings 是外链文件，如下图

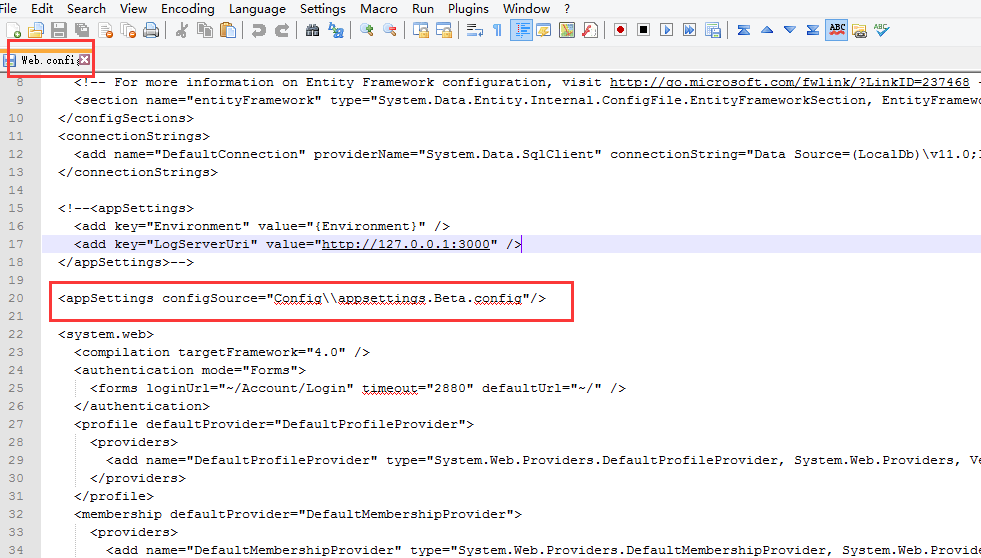


Web.config文件

我们用类似的方法替换(**Replace**)掉这个外链文件路径，如下图



Web.beta.config



发布完成后的web.config文件

Web.Config Transformation具体语法

## 1 ：locator属性

下面有个表，来详细列举locator的语法

**（1）Match;**

这里你需要就是在你直接匹配的属性名。

<connectionStrings>

<add name="Northwind" connectionString="connection string detail"

providerName="System.Data.SqlClient"

xdt:Transform="Replace"

xdt:Locator="Match(name)" />

</connectionStrings>

Engine会再你的Web.config中找到匹配name为Norhwind的就用上面的配置文件图替换。   
**（2）Condition**   
基于XPath,在Locator中应用有逻辑性的判断表达式。

<connectionStrings>

<add name="Northwind"

connectionString="connection string detail"

providerName="System.Data.SqlClient"

xdt:Transform="Replace"

xdt:Locator="Condition(@name=’Northwind or @providerName=' System.Data.SqlClient')" />

</connectionStrings>

上面就是Name属性匹配‘Norhwind’的或providerName匹配System.Data.SqlClient的配置文件节点都会被替换。   
**（3）XPath**   
这个就是直接写XPath, [***http://www.w3.org/TR/xpath***](http://www.w3.org/TR/xpath) ,这里是XPath的标准

<location path="c:\MySite\Admin" >

<system.web xdt:Transform="Replace" xdt:Locator="XPath(//system.web)">

</system.web>

<location>

这里你会发现，这里可以写一些列的表达式。

## 2： Transform 属性

**（1） Replace**   
表示所有匹配的节点都是替换

<assemblies xdt:Transform="Replace">

<add assembly="System.Core, Version=3.5.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089" />

</assemblies>

其实这里描述文件时web.release.config，将要替换的文件时Web.config .   
**(2) Remove**   
删除第一匹配的元素。

<assemblies xdt:Transform="Remove">

</assemblies>

**(3)RemoveAll**

<connectionStrings>

<add xdt:Transform="RemoveAll"/>

</connectionStrings>

**（4）Insert**

插入从父节点中插入，（authorization中插入<deny users="\*" />）

<authorization>

<deny users="\*" xdt:Transform="Insert"/>

</authorization>

**（5）SetAttributes**

<compilation batch="false"

xdt:Transform="SetAttributes(batch)">

</compilation>

**(6)RemoveAttributes**   
删除出Attributes

<compilation xdt:Transform="RemoveAttributes(debug,batch)">

</compilation>

**(7)InsertAfter (XPath)**   
通过匹配 XPath的表达式的，找到节点，并子节点后面插入 XML

<authorization>

<deny users="AName" xdt:Transform="InsertAfter(/configuration/system.web/authorization/ allow[@roles='Admins']") />

</authorization>

**（8）InsertBefore (XPath)**   
通过匹配 XPath的表达式的，找到节点，并子节点前面插入 XML

<authorization>

<allow roles=" Admins" xdt:Transform="InsertBefore(/configuration/system.web/authorization/ deny[@users='\*'])" />

</authorization>

**（9）XSLT (filePath)**

可以在外部定义 XSLT文件，来替换Web.cofig文件。

<appSettings xdt:Transform="XSLT(V:\MyProject\appSettings.xslt)">

</appSettings>

相关链接：

1. <http://www.cnblogs.com/TomXu/archive/2011/11/25/2263089.html>
2. http://www.cnblogs.com/worksguo/archive/2009/08/29/1556307.html