**ASP.NET MVC 4.0分離項目區域**

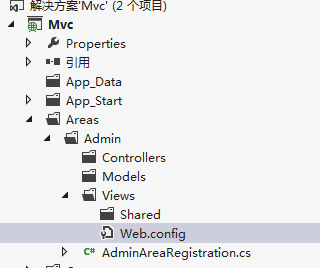
**實現多人開發管理方案**

很曾經開發過ASP.NET MVC 的程序員，在開發大型項目的時候會發現使用MVC框架搭建項目可以很好的吧ADO、ADO Entity、業務層、UI層 劃分開在不同的項目中，視圖層使用ASP.NET MVC去呈現在視圖中。在項目中UI的頁面比較多的時候，會發現需要很多個的區域去劃分UI層的表現，在開發過程中需要多個程序員去編寫項目的UI層。在以前的 ASP.NET MVC 1.0中是無法把UI層劃分出來做成DLL來加載到主顯示程序中來顯示。在2.0 之後就出現可以把每個區域都劃分成不同的項目編譯成不同的DLL加載到主程序上，這樣的做法可以讓開發團隊開更好的共同開發，並且不會產生某個程序員改動了一些樣式或者内部頁面導致其他頁面出現錯誤。

現在我就以一個簡單的ASP.NET MVC 4.0 項目作為例子示範如何把ASP.NET MVC的區域分離成不同的項目。下面需要的框架和擴展插件有: ASP.NET MVC 4.0、NuGet.Tools.vsix、Mvc3Futures、MvcContrib；其中Mvc3Futures、MvcContrib 在VS中的 工具→庫程序包管理器→管理解決方案的NuGet程序包下載直接部署到項目中。

話不說那麽多了，首先我們先安裝好ASP.NET MVC 4.0和NuGet.Tools.vsix，我這裡使用的是VS2012 本身包含了ASP.NET MVC 4.0所以不用下載MVC4.0框架只要下載NuGet.Tools.vsix 插件包就可以了。(NuGet.Tools.vsix 在文章的附件可以下載)。

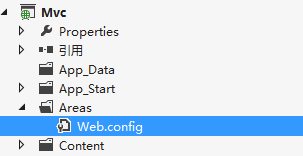
開始動工囖，第一步創建MVC4.0的基本項目在項目右鍵添加區域名稱隨便(本文以Admin為區域名)，反正等下都要刪除。如圖：



這裡我添加了一個區域名叫Admin的

第二步把區域裏面的Views目錄裏面的Web.config 文件移動到 Areas目錄下，然後把整個Admin目錄移除。

結果如圖：



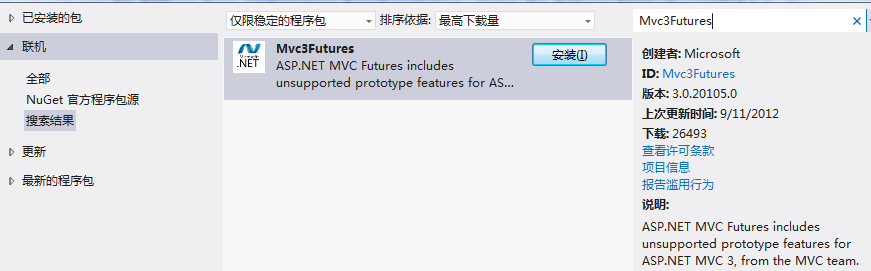
只剩下Web.config 文件

Web.config文件中的内容不用修改

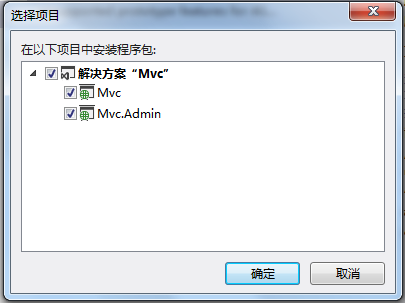
第三步添加多個ASP.NET MVC4.0空白項目**[項目名稱：Mvc.Admin]**，然後添加需要的庫(DLL) Mvc3Futures、MvcContrib 在VS中工具→庫程序包管理器→管理解決方案的NuGet程序包 出現下圖：



點選 聯機 NuGet官方程序包源分別搜索 Mvc3Futures、MvcContrib [題外話：這裡有很多很好的插件大家有空不防去下載來使用] 下面圖片就是部署這兩個DLL的步驟

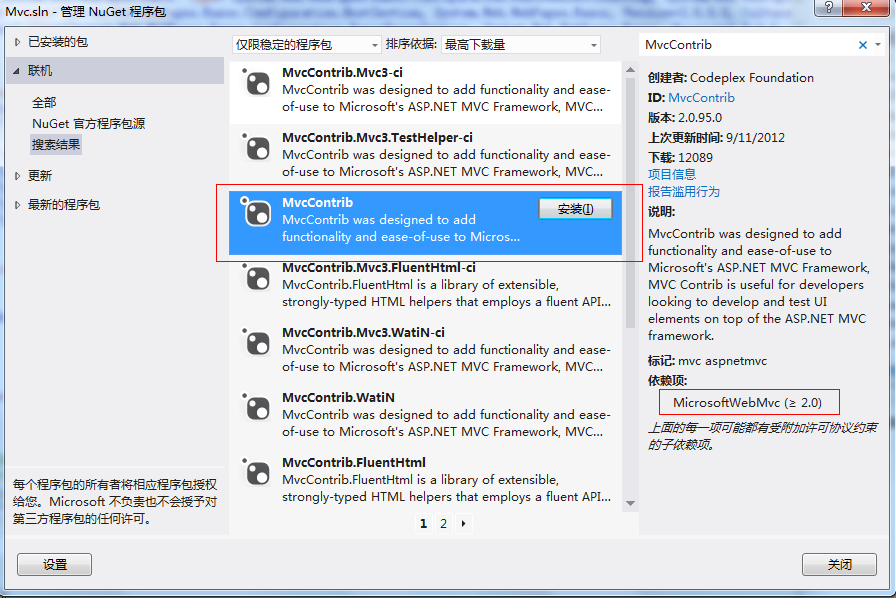


點擊 安裝如下圖

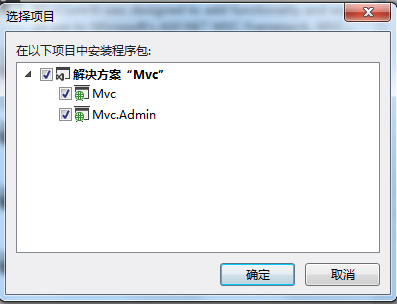


選擇需要部署的項目，點擊確定就會自動部署到需要該DLL的項目中。

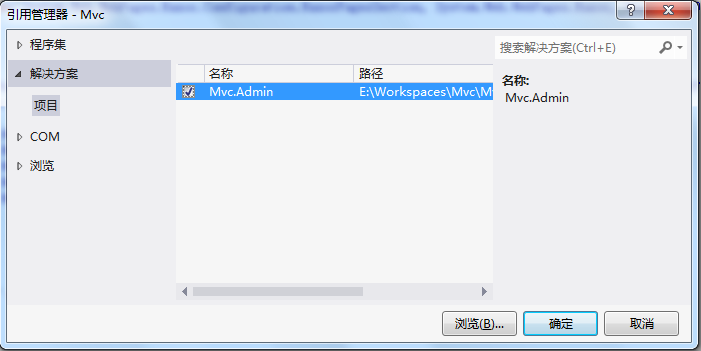
下面繼續安裝部署 MvcContrib 步驟同上如圖:



選擇 MvcContrib 點擊安裝，注意：右面的紅框的位置可以看到該DLL的最低需求。點擊后如圖：

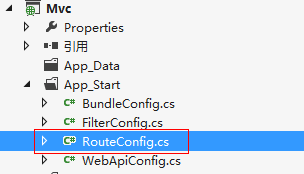


和剛剛一樣，選擇需要部署的項目，點擊確定就會自動部署到需要該DLL的項目中。



好了這些都部署好了，那麽現在就要把他們關聯起來，首先在主項目中引用區域項目如圖：

在VS2012 引用管理器已經改善很多了，速度比以前快好幾倍，加上單選勾非常方便的多選幾個需要引用的項目或者庫(DLL)。

首先我們要改寫一下路由在原基礎上增加命名空間的名字讓程序更好的管理項目調用，在主項目中App\_Start 目錄下的RouteConfig.cs 位子右圖:

點擊打開該文件看見内容中只有一個方法

public static void RegisterRoutes(RouteCollection routes)

在這個方法裏面源代碼

routes.MapRoute(

name: "Default",

url: "{controller}/{action}/{id}",

defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }

);

我只需要修改一下這段代碼就可以，怎麽修改呢只要在參數列中增加關聯的命名空間就可以了

namespaces: new[] { "Mvc.Controllers" } 整合代碼如下：

routes.MapRoute(

name: "Default",

url: "{controller}/{action}/{id}",

defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional },

namespaces: new[] { "Mvc.Controllers" }

);

主項目的設置已經完成跟著下來就是區域項目的設置，之後新增的區域項目后只需要在主項目中添加引用就可以使用了，我們在主項目建立最簡單的控制器和視圖 就是 Home 控制器 和 Index視圖 視圖中隨便寫點東西用來區分不同區域的顯示效果。

下面就是區域項目中的設置，不管3721先把區域項目**[這裡是Mvc.Admin項目]**中 Global.asax 和 App\_Start 目錄下的 文件(FilterConfig.cs、RouteConfig.cs、WebApiConfig.cs) 刪掉 因爲不需要這些了，然後在項目**[Mvc.Admin]**表面添加一個區域組冊的類，這個類有命名規範的必須是 AreaRegistration 結尾的前面名字必須是區域名稱，那麽當前我們就以 Admin為區域名稱 創建類AdminAreaRegistration讓它繼承 MvcContrib.PortableAreas. PortableAreaRegistration

抽象類，主要重寫 public override string AreaName 和 public override void RegisterArea(AreaRegistrationContext context, IApplicationBus bus) 這兩個方法，整個代碼如下:

using System.Linq;

using System.Web;

using System.Web.Mvc;

using MvcContrib.PortableAreas;

namespace Mvc.Admin

{

public class AdminAreaRegistration : PortableAreaRegistration

{

public override string AreaName

{

get

{

return "Admin";

}

}

/// <summary>

/// 标记当前项目的命名空间

/// </summary>

public string AreaControllersNamespace

{

get

{

return "Mvc.Admin.Controllers";

}

}

public override void RegisterArea(AreaRegistrationContext context, IApplicationBus bus)

{

//Resource

if (context.Routes["ResourceRoute"] == null)

{

context.MapRoute(

"ResourceRoute",

base.AreaRoutePrefix + "/resource/{resourceName}",

new { controller = "EmbeddedResource", action = "Index" },

new[] { "MvcContrib.PortableAreas" }

);

}

//Scripts

context.MapRoute(

AreaName + "\_Scripts",

base.AreaRoutePrefix + "/Scripts/{resourceName}",

new { controller = "EmbeddedResource", action = "Index", resourcePath = "Scripts" },

new[] { "MvcContrib.PortableAreas" }

);

//Content

context.MapRoute(

AreaName + "\_Content",

base.AreaRoutePrefix + "/Content/{resourceName}",

new { controller = "EmbeddedResource", action = "Index", resourcePath = "Content" },

new[] { "MvcContrib.PortableAreas" }

);

//Default

context.MapRoute(

AreaName + "\_Default",

AreaName + "/{controller}/{action}/{id}",

new { controller = "Home", action = "Index", id = UrlParameter.Optional },

new[] { AreaControllersNamespace }

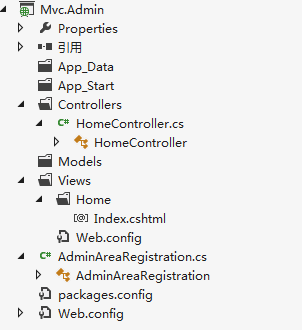
);

RegisterAreaEmbeddedResources();

}

}

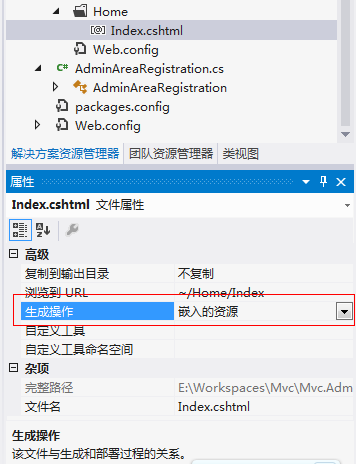
}



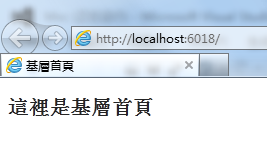
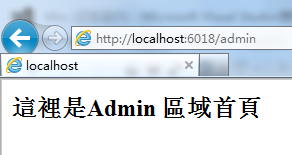
把這段代碼複製到你AdminAreaRegistration.cs文件就可以，這段代碼我優化了一下以後只要更改AreaName 和 AreaControllersNamespace 屬性就可以使用在其他區域項目上

爲了演示在這個區域項目中增加 控制器 Home 和 視圖 Index，在視圖中爲了區分顯示是否在這個區域顯示頁面，我在視圖内容中編寫 ”這裡是Admin 區域首頁” 幾個中文字。項目結構如右圖：

這樣還沒有行我們還差一步，究竟差什麽呢？原來我們的視圖Index.cshtml 在MVC項目是實體文件並非在DLL裏面，我們要設置一下才可以把 Index.cshtml 編譯在DLL裏面，鼠標右鍵點擊 Index.cshtml 文件 屬性，然後把生成操作更改爲”嵌入的资源”這樣就可以把這個文件編譯在DLL中，**如果在開發中有相關的JS、CSS、圖片等文件都需要把生成操作更改爲”嵌入的资源”才會有效。**



好了運行看一下效果

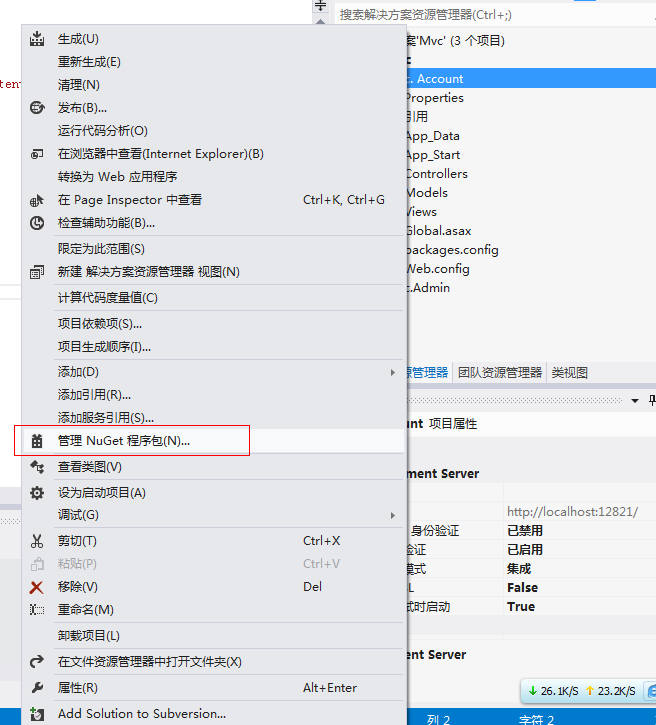
 

運行很成功把項目分離出來了。

下面問題來了如果這個項目擴展多一個區域怎麽去增加呢下面我就示範多個區域項目。

第一步：在解決方案中添加一個新項目[Mvc. Account] 和剛剛上面的區域項目一樣先把不需要的文件刪掉；

第二步：右鍵點擊該項目選擇 管理NuGet程序包 如圖：



然後聯機 安裝 Mvc3Futures、MvcContrib 兩個 DLL ；

第三步：在主項目中添加引用該區域項目；

第四步：添加區域註冊類AccountAreaRegistration.cs 記住要使用命名規範哦！把之前的内容複製過來，修改

AreaName [區域名] 和 AreaControllersNamespace [控制器所在的命名空間] 屬性便可;

第五步：添加控制器和視圖來測試一下 這裡視圖内容是 “這裡是 Account 區域首頁”;

第六部：運行測試結果如下圖。

