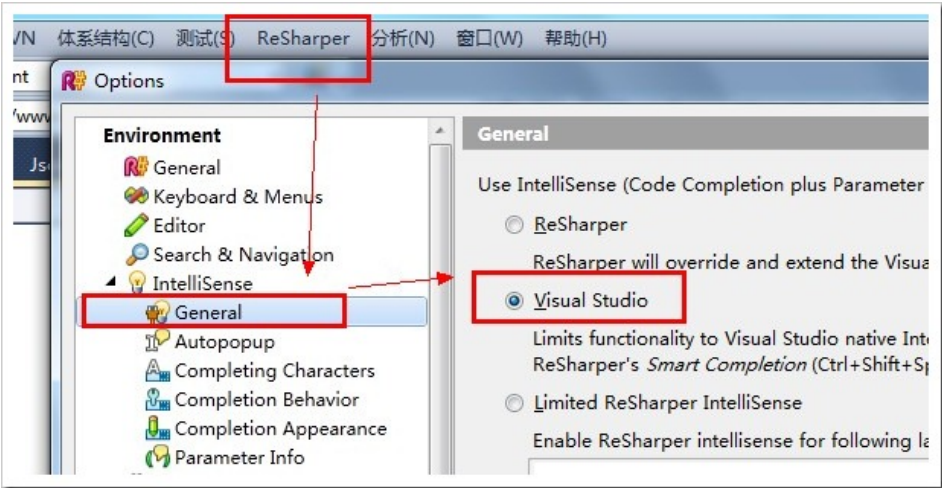
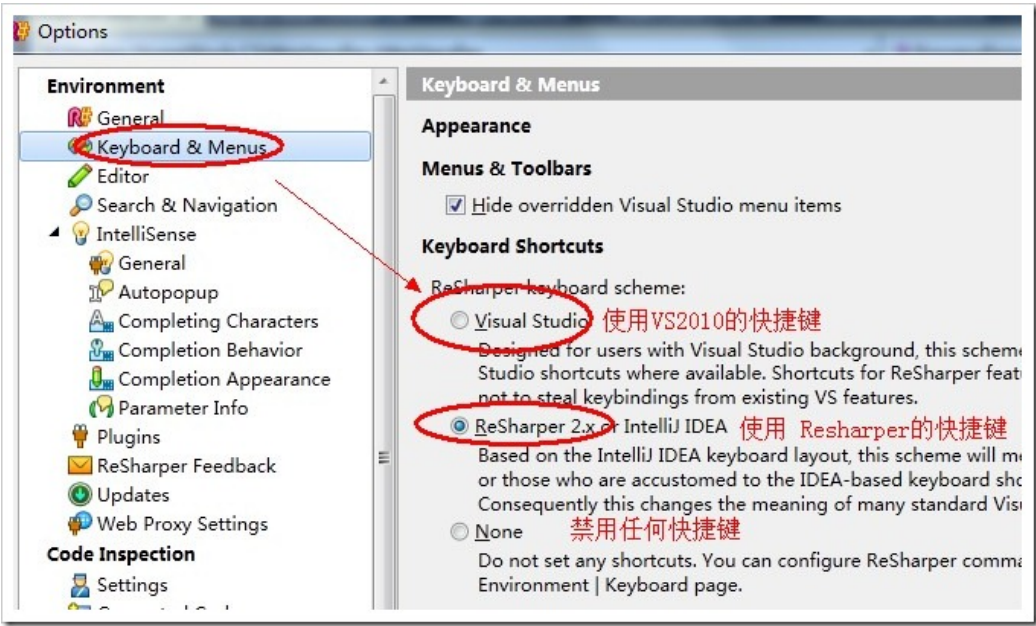


ReSharper 配置及用法(转)

1:安装后，Resharper会用他自己的英文智能提示，替换掉 vs2010的智能提示，所以我们要换回到 vs2010的智能提示



2:快捷键。是使用vs2010的快捷键还是使用 ReSharper的快捷键呢？我是使用re的快捷键



3:

esharper安装后，会做几件事情，这几件事情对于除此使用者，比较麻烦，因此归纳总结一下，以资参考。

(1)、会将选项——文本编辑器——C#——常规——自动列出成员 这个选择框的勾选去掉。这样当你使用某个方法的时候，便不会提示参数，是一个很郁闷的事情。

可以手动勾选。

(2)、会将选项——文本编辑器——C#——高级——显示实时语义错误、在编辑中用下划线标识错误这两个选项去掉。这个本来没有什么，但是如果卸载了Resharper，

vs编辑器变不会实时提示错误。因此卸载Resharper后，需要把这两个选项重新勾选上。

昵称: [vingi\\_苍月](#)  
园龄: 7年1个月  
粉丝: 36  
关注: 5  
[+加关注](#)

< 2016年12月 >

日	一	二	三	四	五	六
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

搜索

找找看

常用链接

[我的随笔](#)  
[我的评论](#)  
[我的参与](#)  
[最新评论](#)  
[我的标签](#)

我的标签

linux(1)

文章分类(110)

[C#\(45\)](#)  
[CSS\(7\)](#)  
[JS\(10\)](#)  
[Linux\(9\)](#)  
[MariaDB\(2\)](#)  
[NoSQL\(5\)](#)  
[Other\(8\)](#)  
[Python\(16\)](#)  
[Ruby\(3\)](#)  
[SQL\(5\)](#)

最新评论

1. Re:检测到在集成的托管管道模式下不适用的ASP.NET设置的解决方法(转)  
这个问题困扰我一段时间了，今天终于解惑了，感谢楼主，大赞  
--小离子
2. Re:C#使用DotRas完成ADSL自动重连

4: use var

use implicitly typed local variable declaration

楼主，可以实现动态更新账号和密码吗？  
--OK121

Options

Environment

General

Keyboard & Menus

Editor

Search & Navigation

IntelliSense

Plugins

ReSharper Feedback

Updates

Web Proxy Settings

Code Inspection

Settings

Generated Code

Inspection Severity

Custom Patterns

Code Annotations

Code Editing

Code Style Sharing

General Formatter Style

Members Generation

File Header Text

Code Cleanup

Context Actions

Localization

C#

Visual Basic .NET

HTML

ASP.NET

Razor

JavaScript

CSS

XML

Inspection Severity

Search

AllC#VB.NETCSSJSHttpHandler or WebServiceASPXXAMLHTML

Class cannot be instantiated

Convert anonymous method to method group

Convert 'if' statement to 'switch' statement

Convert 'Nullable<T>' to 'T?'

Convert property to auto-property

Convert property to auto-property with private setter

Convert static method invocation to extension method call

Convert to lambda expression

Convert to static class

For-loop can be converted into foreach-loop

'if' statement can be re-written as '?.' expression

'if' statement can be re-written as '??' expression

'if-return' statement can be re-written as 'return' statement

Introduce optional parameters

Invert 'if' statement to reduce nesting

Loop can be converted into LINQ-expression

Part of loop's body can be converted into LINQ-expression

Use object or collection initializer when possible

Use 'var' keyword when initializer explicitly declares type

Use 'var' keyword when possible

3. Re:检测到在集成的托管管道模式下不适用的ASP.NET设置的解决方法(转)

抓住了问题的本质

--hollyspirit

4. Re:C#使用DotRas完成ADSL自动重连

解决大问题，这个写法才是正确的，网上的其他写法都会提示“系统无法为这个连接找到电话簿项目”

--郭明锋

5. Re:sublime text 3 plugin LiveReload的应用 for chrome

html页面必须放在服务器下面吗

--梦想.家

C#	VB.NET	Suggestion
	C#	Suggestion
	C#	Suggestion
	C#	Suggestion
	C#	Suggestion
	C#	Suggestion
	C#	Hint
	C#	Suggestion
	C#	Hint
	C# VB.NET	Suggestion
	C# VB.NET	Hint
	C#	Suggestion
	C#	Do not show
	C#	Do not show

5:按 alt+enter 能解决很多问题。下面是快捷键大全。



## 7:威力无比的Alt+Enter回车

万能的Alt+Enter能够帮你完成很多编写代码过程中的dirty work，总结起来大概是这么些：

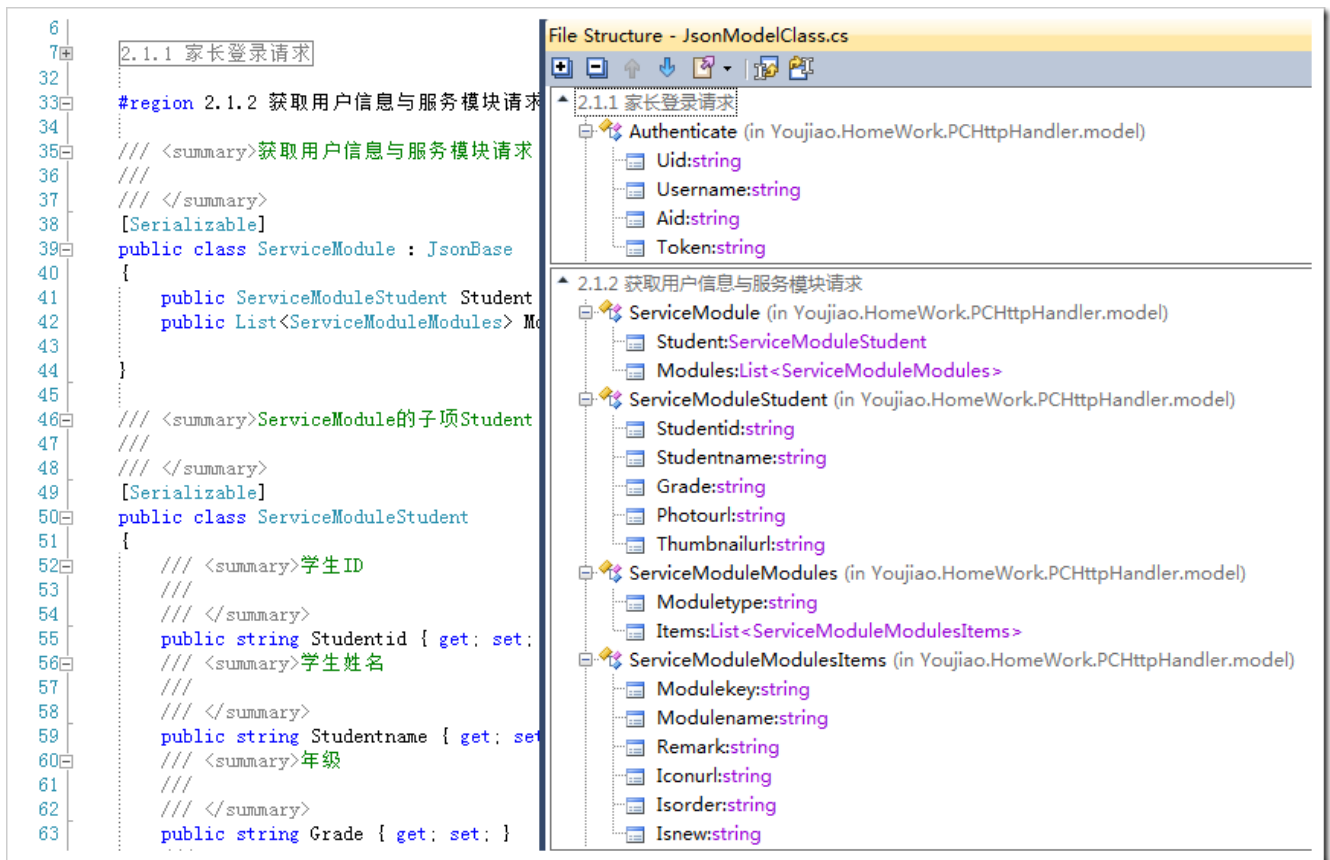
- 帮你实现某个接口或抽象基类的方法；
- 提供你处理当前警告的一些建议；
- 为你提供处理当前错误的一些建议（不一定是真的错误）；
- 为你简化当前的臃肿代码；

**8: Ctrl + F11** 当我们看别人的代码，或者是看自己的代码的时候，总是觉得代码太多，于是我们就用region来把代码进行了封装注释，可是这样之后别人看代码就很郁闷，Resharper的 File Structure 功能，就可以把region和你的方法都展示出来。

说了这么多，其实就是把对象浏览器和region的长处结合起来，既可以清晰的分类，又能一目了然的找到需要的方法。Resharper这时帮上你的大忙了。用Ctrl+F11，就弹出一个像右边这样的窗口来。

这里面，按照你的region来显示，这样读你的代码的人也受益了。每个方法的参数，返回值都如UML一样列出来。

- 如果需要浏览到某个方法，直接双击它的名字；
- 如果要把某几个方法装进一个新的region，则可以选中方法，点工具栏上的像框的那个图标；点叉则会删除这个region并把相应的方法移到外面来。
- 如果要调整某个方法的位置，比如把它移到别的region里面去，只需要在这里拖动这个方法即可。
- 更可喜的是，你想要的从这里浏览、找到所有使用 and 重构的功能也在这里提供了，在某个方法上右键你就能开始操作。



## 9:重构才是王道（上）

重构是一种精神，证明你在致力于提供高效的、精炼的、健壮的代码，而不是凌乱的、晦涩的、漏洞百出的代码。

在Visual Studio 2005中，微软第一次提供了重构工具。但是不够，远远不够。我们需要的重构是非常广义的，我们想要对代码进行快速的调整，快到我在我想什么我的工具就能做什么。这才是追求重构的境界。所以在这个意义上，几乎Resharper为你提供了巨大的生产力。

Visual Studio 2005提供的重构包括了如下：

1. 封装字段
2. 提取方法

3. 提取接口
4. 提升局部变量
5. 移除参数
6. 重命名
7. 重新排列参数

这些方法在Resharper中全部都支持（但Resharper的重构远不止这些），它们对应的变成了：

1. 封装字段 —— Introduce Field
2. 提取方法 —— Extract Method
3. 提取接口 —— Extract Interface （另增加了Extract Superclass提取为基类）
4. 提升局部变量 —— Introduce Variable
5. 移除参数 —— 移到Change Signature（改变方法签名）中
6. 重命名 —— Rename （Resharper会根据对象的类型名称，提供你几个可选的最合适的名称）
7. 重新排列参数 —— 移到Change Signature（改变方法签名）中

我知道很多人都声称自己E文不好，但是，这确实都是很简单的单词，难不倒任何人的。这些重构的功能是人所共知的，下面就告诉大家一些Resharper特有的，首先，重构的快捷键是Ctrl+Shift+R：

1、对于类，除了提取接口、基类，你还可以移动它到其他的命名空间和移动到别的文件里，这是一个实用的功能，也许你不信，但是我这真的有人，把所有的business entity都写在一个DataObject.cs里面。你难以想象，我打开它时嘴张了多大。

2、对于字段，提供了：

- Safe Delete，会检测所有使用到的地方，并询问如何删除；
- Pull Member Up和Push Member Down，可以把这个字段在基类和继承类中移动；
- Use base type where possible，尽可能的使用基类，由于ArcGIS平台是基于com组件的，很多时候我们需要的是IGeometry, IPointCollection这样的接口所公开的属性或者方法，于是你没必要保存一个polygon对象，而可以使用基类型；
- Encapsulate Field，封装字段，但是这个功能远没有另一个提供同样功能的操作有用。我可以在后文中来讲。

3、对于方法，提供了：

- 与字段类似的功能，此外；
- Change Signature，更改函数签名，包括更改名称，返回值类型，参数的各种信息，添加和删除参数，相当实用。如果你是在重写方法上操作，会提示你是否到基类中更改。
- Make Static，如果Resharper检测到这个方法并没有与非静态成员相关联的话，往往会自动地提示你（以黄色横杠的形式出现）可以改为static，如果你自作主张的对一些方法进行修改也无不妥，但后果自负。
- Extract class from parameter，如果你的参数有七个八个，那是否考虑用一个类来封装这些参数呢，于是这个功能应运而生。
- Method to Property，顾名思义，如果还在使用GetField()或者SetField(..)的话，你一定是从非.net星来的。

4、在方法体内部：

Extract Method，不用介绍了吧。

Introduce Variable/Parameter/Field，取决于你光标所在的对象，可以提供转化的功能。

Inline Variable：就是把：

```
IPoint point = new PointClass();
point.PutCoords(_point.X, _point.Y);
```

变成这样子：

```
new PointClass().PutCoords(_point.X, _point.Y); //这是个糟糕的例子
```

5、重命名：

为什么重命名值得挑出来讲，因为Resharper提供了命名建议这一金子般的功能。于是，想改名为易读性强的名字，不是那么费脑子的事情了。Resharper会根据这个变量的类型，为你提供几个备选名字，名字列表是列在光标位置上的（对方法重命名会弹出对话框），你只需要用方向键选择并敲回车即可，这种名字多是将类型的名字首字母改为小写得来的，甚至刨根到基类的类型名，你还可以在此基础上加以改进。如果你还在用i,j这种晦涩的名称，请



迅速的把他们改为outIndex, pointCount之类可读的名称。

Resharper其实提供了更先进的功能，在你命名一个变量时，就有快捷键为你提供备选名字，但是Ctrl+Space是我们宝贵的输入法切换键，于是，我对变量名的敲定，往往是先起了一个较烂的，然后重命名的。

还有一些更广义上的，帮助你代码进行调整的功能，我另写一篇吧，不然太长了。

重构才是王道（下）

#### 插入代码

Resharper的Alt+Insert快捷键提供给你插入代码的功能。由于这两个键非常难按（这是我的感受），真正在使用的时候，我用的是Alt-R-C-G，意指打开Resharper菜单——Code——Generate，都只需要你的左手，这样你可以右手一边比划，一边还在写代码，多酷啊。

- 生成的代码中最常用的是构造函数和属性，当你没有私有字段的时候，只会生成一个空的默认构造函数，而且没有生成属性的功能。在你有私有字段的情况下，生成之前会让你选择哪些私有字段需要作为构造函数的参数，并生成初始化的代码，这样编写重载极其方便。生成属性也类似。
- 再次常用的就是重写基类或者接口的方法了。选择Implement Interface Member或者Override Inheritate Member，Resharper会查找当前类的基类或接口，然后按继承层次列出来，根据你的选择重写或实现这些方法。
- 不是太常用的是生成Equals和GetHashCode方法，在我的应用场景中很少重写它们。但是根据《.NET设计规范》，不管是值类型还是引用类型的Equals都建议重写，并且应该重写GetHaseCode方法，因为它们相互依赖。如果你有这个需求，那么生成这三个函数一定能够帮你的大忙。

#### 包围代码

Visual Studio也提供了外侧代码这个功能，你可以按Ctrl+K, Ctrl+S来激活这个功能，虽然我并没有任何鄙视Visual Studio的意思，但是Resharper的快捷键确实更加合理（我在按下Ctrl的时候真的很难按下S），条目也更加清晰。Resharper中这个功能的快捷键是Ctrl+Alt+J，然后你就可以选择将当前行的代码包围到try-catch块或者using中了。这是很高效的方法，我们倾向于在开发的早期尽量不捕获异常，而在中后期才加入异常处理机制。于是你某一个时期有大量的工作是把他们扩到try-catch块中。而你要使用支持dispose对象时，最好的方法是使用using块。（卖蛋糕的，当我知道我的代码不是最优的时候，我总是寝食难安），这里自然也有把代码扩到region块中的功能，也是常用功能之一。

#### 调整方法的位置

前面我曾说过，如果要调整方法的位置，可以在代码结构窗口中拖放操作。如果你觉得只是把一个方法移动到前面去，却不得不打开代码结构窗口太过重量级，那么有轻量级的方法：当光标位于方法的名称上时，用Ctrl+Shift+上下键就可以移动方法的位置，包括方法的xml注释，但如果你用的不是三个/的xml注释而是两个/的，那么就对不起不了。

#### 其他琐碎的功能

你肯定常常会复制粘贴当前行的代码，例如在使用StringBuilder.Append的时候，Ctrl+D可以简化你Ctrl+C, Ctrl+V的工作。

曾经有一个组合键可以注释掉当前行，还有另一个是取消注释，但是我已经淡忘了，因为Ctrl+/才应该是真正属于它的快捷键，再次按下就可以取消注释。

关于Resharper的重构功能就是这些，我可能天真地把很多额外功能都算在重构里了，但是它确实能够帮助你快速的对代码进行调整和优化。所以，请不要深究我对重构的概念认识是不是混乱。

增强的浏览功能

#### 浏览参数的方式

输入方法的时候，我们已经习惯了由IDE提供给我们的参数提示，极大方便了我们的选择重载方法。在没有Resharper的环境下，Visual Studio已经做到了。那么为什么Resharper还要增强这个功能并大获好评的。试问，Visual Studio那窄窄的一行参数提示有没有让你觉得憋屈。我们有19寸的大屏幕，1600的分辨率，却不得不盯着那窄条条，小心翼翼的按着上下键寻找我们需要的重载。至少，开发Resharper的伙计是受不了这种憋屈的，于是大开大阖版的参数列表出现了，长长的参数重载被以列表的形式展现出来，当你在使用GDT+方法，看到巨大的参数重载时，你会从心底里发出感叹：卖蛋糕的。

同时，Resharper展示参数的快捷键变成了Ctrl+P，如果你觉得屏蔽了打印的快捷键简直是在开玩笑的话，那么问问你自己有多少次打印过自己的代码。

#### 浏览打开过的文档

我窃以为你已经知道了在Visual Studio中切换文档的方式，它们包括：

Ctrl+Alt+上下方向键，可以在打开的文档中切换；

Ctrl+Tab，不仅可以在文档之前切换，并可以切换到解决方案文件夹，属性视图去，需要按左右键。

但是怎么样打开最近编辑后关闭的文件呢，Visual Studio很客气的又没有提供此功能，于是留给了Resharper。在我这里这个快捷键是Ctrl+E,Ctrl+E，没错，按两次。如果你的不是，那么在Resharper-View-Recent Files菜单下看看它是什么。因为你会时常用到。打开一个文件的列表，用方向键选择并回车就会在编辑器中打开。

很多人说Resharper的性能问题，我想，一个可能的原因是打开的文档太多了，如果你有时刻关闭不需要的文档的习惯，性能或许不会那么差，并且你可以随时打开这些你关闭了的文档，就像在已经打开的文档中切换一样的方便。

我的团队中没有用到敏捷开发那些高级的东西，但是我们还是保持着每次改动都仅涉及两三个文件的好习惯，并且频繁的commite到源代码服务器上去。所以，我每次真正要编辑的文件不多，性能不是问题。

和大家分享了很多Resharper使用的技巧，点点滴滴都已经融入我日常的开发工作中了。当然很不全面，例如与Nunit的集成，由于我们不是测试驱动，所以没有任何体验，自然也不敢大放厥词；也很主观，我觉得它好，你可能觉得它不好，萝卜青菜各有所爱。再说，它也不是没有白痴的地方，在文档上点右键增加的一个Close All功能，可以关闭所有打开的文档，关闭了干什么，对着一个空白的屏幕发呆么？我觉得原生的“除此之外全部关闭”就够了。还有一个定位的功能（Locate in Solution Explorer），真是没用，如果你在VS选项中设置了，在解决方案管理器中跟踪活动项，那么VS自动就给你定位了。

不管怎么说，它带给我更快更方便的开发体验，把我从一些琐碎的，不人性化的功能中解放出来。从这一点上来说，我很希望越来越多的人喜欢上它，开始用它，并帮助它更好的发展。

分类: C#

好文要顶

关注我

收藏该文

vingi\_苍月  
关注 - 5  
粉丝 - 36  
[+ 加关注](#)

21

1

posted on 2013-01-29 16:57 [vingi\\_苍月](#) 阅读(25344) 评论(11) [编辑](#) [收藏](#)

评论

- dubuyunjie

| 2013-06-29 21:52

# 1楼

请问博主我的sharper的重命名是灰色的，怎么办啊

支持(0) 反对(0)
- Claude He

| 2013-08-22 12:37

# 2楼

转载请提供原文出处！

支持(0) 反对(0)
- 陈大欠

| 2013-10-24 14:32

# 3楼

很棒~

支持(0) 反对(0)
- The Mechanic

| 2013-12-25 18:10

# 4楼

LZ,  
5:按 alt+enter 能解决很多问题。下面是快捷键大全。  
下面的图使用什么软件做的？

支持(0) 反对(0)
- 6572789

| 2014-02-21 16:25

# 5楼

不错

支持(0) 反对(0)
- czcz1024

| 2014-03-06 10:06

# 6楼

我还真觉得Locate in Solution Explorer是个好功能  
因为我默认是关闭 在解决方案管理器中跟踪活动项  
个人习惯不一样

支持(0) 反对(0)

→ [liudegui](#) | 2014-03-12 10:39

#7楼

从哪儿转的？

支持(0) 反对(0)

→ [路正遥](#) | 2014-03-17 16:48

#8楼

R#的配置还是比较难找的，个人更倾向于通过xml配置，还没有找到相关的文档。。

支持(0) 反对(0)

→ [kuibono](#) | 2014-03-25 09:08

#9楼

@ 瓦罐  
看样子是 xmind

支持(0) 反对(0)

→ [VAllen](#) | 2014-08-13 16:33

#10楼

Mark

支持(0) 反对(0)

→ [VAllen](#) | 2014-08-14 09:07

#11楼

为什么要取消var的提示？

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#)网站首页。

最新IT新闻：

- [Crytek关闭多处工作室](#)
- [6770元！高级果粉必备神器UltraFine 5K显示器开卖](#)
- [微博一日不死，其余付费问答产品都是太子](#)
- [告别蓝牙！大屏幕时代，设备连接，一“触”即发](#)
- [被MD5加密坑了？雅虎大规模泄密是个咋样的悲剧](#)
- » [更多新闻...](#)

最新知识库文章：

- [写给未来的程序媛](#)
- [高质量的工程代码为什么难写](#)
- [循序渐进地代码重构](#)
- [技术的正宗与野路子](#)
- [陈皓：什么是工程师文化？](#)
- » [更多知识库文章...](#)