StackOverflow：

QT 插件不易用，查找定义位置有时候会出错

Any disadvantages of using Qt visual studio addon [closed]

Question：I am just starting to learn how to use Qt in C++. They have a option to use a standalone Qt creator as well as an addon for Visual Studio. I have Visual Studio Community 2015.
The question is, is there any disadvantages of using the addon for Visual Studio over just coding in Qt creator?

Answer：*It seems the below is a bit outdated already considering using Qt extension for VS2015. But I have not personally tried myself yet and suspect Qt Creator still have its advantage as below.*
The advantages of Qt Visual Studio Add-In:

- It lets us to use better debugger from Visual Studio while working with Qt project
- It lets us to easily do the remote debugging by running the executable on other system or VM
- Many developers used to Visual Studio

The disadvantages of Qt Visual Studio Add-In:

- Very unreliable project import, makes us to fix things manually. And not all add-in versions are good: we still use 1.2.3 because of that
- Not convenient with adding file to the project etc. We usually add to and then re-import the project from Qt .pro file
- Visual Studio builds the code much slower than Qt Creator does (roughly the proportion between nmake and jom tools, jom parallelizes the build)
- Visual Studio must be higher than Express (e.g. Ultimate)

The rest of advantages and disadvantages can be considered individual developer preferences. I like Qt Creator for its ability to quickly find any definition under the cursor (press Ctrl and hover the mouse cursor and select) while MS Visual Studio does that but not as easy and surprisingly too frequently fails to find the definition (they fix it for years and maintain huge index files for the source code while Qt Creator does not build such files).
And for me the main disadvantage of Qt Creator is CDB from Windows SDK or maybe another external one vs. way better debugger in Visual Studio including remote. So I debug complex issues with Visual Studio but I write the code and quickly test it with Qt Creator.

visual studio 的快捷键

Command to collapse all sections of

code?

In Visual Studio (2008) is there a command to collapse/expand all the sections of code in a file?

CTRL + M + O will collapse all.

CTRL + M + L will expand all. (in VS 2013 - Toggle All outlining)

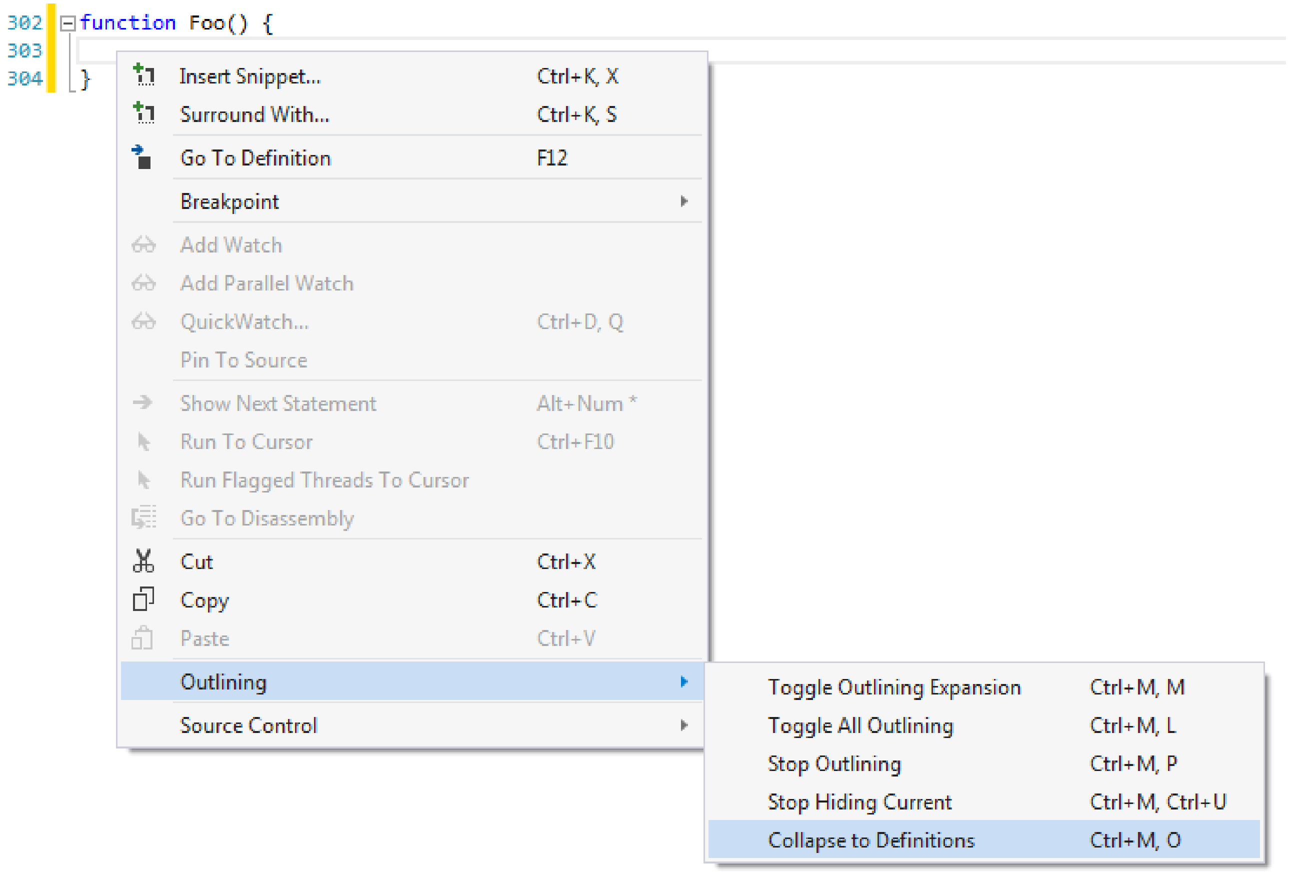
CTRL + M + P will expand all and disable outlining.

CTRL + M + M will collapse/expand the current section.

CTRL + M + A will collapse all even in Html files.

These options are also in the context menu under Outlining.

Right click in editor -> Outlining to find all options. (*After disabling outlining, use same steps to enable outlining.*)



```
function Foo() {

}
```

Insert Snippet... Ctrl+K, X
Surround With... Ctrl+K, S
Go To Definition F12
Breakpoint
Add Watch
Add Parallel Watch
QuickWatch... Ctrl+D, Q
Pin To Source
Show Next Statement Alt+Num *
Run To Cursor Ctrl+F10
Run Flagged Threads To Cursor
Go To Disassembly
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Outlining
Source Control
Toggle Outlining Expansion Ctrl+M, M
Toggle All Outlining Ctrl+M, L
Stop Outlining Ctrl+M, P
Stop Hiding Current Ctrl+M, Ctrl+U
Collapse to Definitions Ctrl+M, O

无法使用 Git，Git 插件不易用

Using Git with Visual Studio [closed]

As a long-time Visual SourceSafe user (and hater) I was discussing switching to SVN with a colleague; he suggested using Git instead. Since, apparently, it can be used as peer-to-peer without a central server (we are a 3-developer team).

I have not been able to find anything about tools that integrate Git with Visual Studio, though - does such a thing exist?

What are the technologies available for using Git with Visual Studio? And what do I need to know about how they differ before I begin?

快捷键规范化代码

How do you auto format code in Visual Studio?

I know Visual Studio can auto format to make my methods and loops indented properly, but I cannot find the setting.

To format a selection: Ctrl+K, Ctrl+F
To format a document: Ctrl+K, Ctrl+D

See the pre-defined keyboard shortcuts. (These two are Edit.FormatSelection and Edit.FormatDocument.)

自定义 Visual Studio 编辑器中的替代字体，操作复杂，对初学者不易用
Overriding font in custom Visual Studio editor

The problem is in making custom editor inside VS extension look differently than the current theme dictates. The editor is hosted inside a dialog and should have the same font the hosting dialog defines.

The content type of the editor is defined like this:

```
[Export]
[Name("MyContent")]
[BaseDefinition("code")]
public static readonly ContentTypeDefinition ExportContentTypeDefinition = null;
```

And there is a classification type definition:

```
[Export]
[Name("MyContentText")]
[BaseDefinition("text")]
public static readonly ClassificationTypeDefinition MyTextDefinition = null;
```

The classifier provider is defined as below:

```
[Export(typeof(IClassifierProvider))]
[ContentType("MyContent")]
public class ClassifierProvider : IClassifierProvider
{
    [Import]
    public IClassificationTypeRegistryService ClassificationTypesRegistry { get; set; }

    public IClassifier GetClassifier(ITextBuffer textBuffer)
    {
        return new Classifier(
            ClassificationTypesRegistry.GetClassificationType("MyContentText"));
    }
}
```
While the classifier just provides the same format for any snapshot:

```
public class Classifier : IClassifier
{
    private readonly IClassificationType _classificationType;

    public Classifier(IClassificationType classificationType)
    {
        _classificationType = classificationType;
    }

    public IList<ClassificationSpan> GetClassificationSpans(SnapshotSpan span)
    {
        return new [] { new ClassificationSpan(span, _classificationType)};
    }

    public event EventHandler<ClassificationChangedEventArgs> ClassificationChanged;
}
```
Now, in code, while creating the editor, I'm trying to override the properties of the matching `IClassificationFormatMap`:

```
var contentType = contentTypeRegistryService.GetContentType("MyContent");
var textBuffer = textBufferFactoryService.CreateTextBuffer(initialText, contentType);
var textView = textEditorFactoryService.CreateTextView(textBuffer);


...


var formatMap = classificationFomatMapService
    .GetClassificationFormatMap("MyContentText");

formatMap.DefaultTextProperties = formatMap.DefaultTextProperties
    .SetFontRenderingEmSize(dialog.FontSize)
    .SetTypeface(
        new Typeface(
            dialog.FontFamily,
            dialog.FontStyle,
            dialog.FontWeight,
            dialog.FontStretch));
```

However, the change doesn't affect my editor instance.

Moreover, the format map returned from the `classificationFomatMapService.GetClassificationFormatMap(ITextView)` overload is different from the one returned from the overload I use above. And changing this another instance of format also affects all the code editors in the running Visual Studio instance, so I have to conclude that despite my efforts the *textView* somehow maps to the default editor's classification.
My question is: what should I do in order to control text appearance of a custom editor designated for a custom content type?

VS 会格式化 HTML 标签和内联代码，因此它非常昂贵

如果在源代码视图中并触摸“属性”面板窗口以更改控件的属性，则可能会从页面中删除所有事件连接

Re: disadvantages of using visual studio?

It changes your HTML in the ASPX page. It occassionally crashes.

Yeah!, if you're anal about the spacing/layout of your html source be forewarned. The design mode has tons of unfixed bugs (but should be fixed with VS 2005). It messes up your html source code by reformatting it how it thinks it should be, and doesn't allow you a way to turn this off. Only way around it is to not use design mode, or catch it in the act and hit ctrl-z to undo.

once you work on VS then you won't use any other editor/tool but its very expensive watch out for HTML and design view because VS formats HTML tags and inline code. But **VS provides a complete environment for developers**

- I'll add, once you work on VS.net you almost can't use any other editor/tool even if you wanted to. Well, you could, but it would be hard. VS.net also only supports code-behind. You can write single page aspx page, but it's not supported. The worse are the bugs as mentioned in previous posts. Also if you are in source code view and touch the properites panel window to change a control's properties you run the chance of having all your event hookups erased from the page. You won't notice this until your page no longer works. This is a pretty old bug, so I'm not sure if it was fixed. I don't like how it has to create projects (and tons of files not needed for production) for an ASP.net site, but I don't know if I'd call that a disadvantage.

- Another bug I've come across recently is when you try to share common controls/methods in a base class that your codebehind class inherits. It really shows the bugs in the design time compiler, I'll start getting errors of "cannot find sub blahblah()" when sub blahblah() is right above the error. If I close Visual studio and reopen my project, the error goes away and everything works great again. That is getting really annoying...

- One of the problems that I have come across is that VS.net seems to want to force you into Microsoft's pattern of solutions and projects using an isolated development model. I had the hardest time converting a development project from a shared enviroment on a live remote server to be able to us VS.net. I eventually gave up with it. My advice is that you either work totally with VS.Net and follow their developoment philosophy or use something else. As with most of the other complaints this problem will also be solved in the next version of Vs.net

The "design-view-munging-HTML" issue will be fixed fully in the next version. The next version will also have support for intellisense in single file aspx pages. The project model is very much different than VS 2002 and VS 2003. You can basically point at a location and open it as a web. The disappearing event handler bug is also gone. Besides these obviously huge issues that we're already addressing in the next version, are there other things you would like to see that would make you never want to use another tool for editing your aspx webs? This would be good feedback for us to get. Thanks, -- Bash

Quaro:

Microsoft Visual Studio 过于复杂，导致对初学者不友好

Microsoft Visual Studio for introduction to programming?

I don’t think it’s good for beginner programming lesson. It is like learning to drive with a Boeing jumbo-jet. Don’t get me wrong, Visual Studio is a fantastic tool, it is awesome tool for professional engineers who need shortcuts, productivity boosts, and all-rounded Application Lifecycle Management tools.

But “click this button, click that button, automatically generate a method here, run stress test, etc.” definitely are **not** good ways to teach programming.

In my opinion **no**.

For the record, I think Visual Studio is an awesome tool. You simply won't find a more powerful IDE. But it's designed to make professional programmers more effective, not to provide the best possible learning experience.

The problem is that Visual Studio holds your hand just a little bit too much. Between the autocomplete, IntelliSense and auto-generated boilerplate code, the programmer is left only to think about how to solve the problem, not how to write the code. If you install ReSharper, you hardly have to think about refactoring either.

If I create a new Windows Forms application in Visual Studio, my Program.cs file already looks like this:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    static class Program
           {
              /// <summary>
              /// The main entry point for the application.
              /// </summary>
              [STAThread]
              static void Main()
              {
                 Application.EnableVisualStyles();
Application.SetCompatibleTextRenderingDefault(false);
                 Application.Run(new Form1());
              }
           }
        }
```

For someone trying to learn, this is not good. You should ideally be typing in the using statements, the namespace, the class and the Main method manually, and make sure that you understand what every line is doing.

The best way to learn programming in my opinion is like this: You start by writing commands into an interactive shell (an REPL). Your first exposure to programming would then be something like this (this is Python):

```
>>> 2+2
```

```
4
>>> print("Hello, World!")
Hello, World!
```

Once you've mastered a few basic commands, you open a *very simple* code editor and start writing your first program, line by line.

For these reasons, I think Python would be a better programming language for an introductory course. Java has traditionally been the most popular language in these courses, but it also has many of the same problems as C#.

CSDN:

界面设计没有充分考虑空间占用

Visual Studio 的工具窗口占用了过多的空间。ErrorList 在标题栏和 Tab 标签出现了两次，实际上这是无意义的重复，而标题栏中央和工具栏右边的部分，有大片地方被白白浪费掉了。在宽屏笔记本上，屏幕高度相当有限，这一点格外令人难以容忍。

Eclipse 的窗口对空间的利用有效得多。标签页和按钮共享同一行，并且因为 Eclipse 的错误显示是分类的，也不需要三个额外的按钮，所以界面相当紧凑。

输出显示过于单调

Visual Studio 的输出窗口只有一种颜色、同样的格式，在密密麻麻的输出里面根本分不清重点。Eclipse 的输出窗口能显示多种颜色和格式，能够清楚的分出不同的内容，和代码关联的内容还可以用 URL 导航。

跟踪活动项不方便

Visual Studio 有这样一个功能叫做 Tracking Active Item，有时候我需要它，有时候又希望关闭它。Visual Studio 把它放在选项里面，每次修改的时候都要重复点菜单-》选分类-》点 CheckBox-》点 OK 这样一个重复的动作。

Eclipse 把这个功能叫做 Link with Editor，放在主界面的 Package Explorer 的工具按钮里，打开和关闭都只需要点击一次鼠标就行了。

设置代码格式不够灵活

看上去 Visual Studio 也提供了不少代码格式选项，可还是缺少很多高级设置，比起 Eclipse 只能算小巫见大巫了。比如，我很喜欢 Eclipse 把字段对齐这个功能，但 Visual Studio 压根没有这个选项。

代码行定位功能不如 Eclipse

Eclipse 编辑器右侧有个特殊区域代表了整个代码文件，断点、错误、警告、书签都会在这里标识出来，不论代码有多长，点击一下就可以定位，非常的方便。

在 Visual Studio 里面定位代码，要么用鼠标滚轮上下翻动直到找到内容，要么从成员列表的组合框里选择，不论哪一种，都没有 Eclipse 来得简洁方便。

设置文件编码太过麻烦

Visual Studio 既不能指定文件的默认编码，也不能批量设置文件编码，只能通过 Save As 对话框下面一个很小的箭头手工一个一个指定，麻烦到文件多的时候我有一种想砸了它的冲动。

Eclipse 可以从文件类型、工作区、项目、单个文件四个级别设置文件编码，而且允许手工输入编码名称，这比 Visual Studio 从一个长长的列表里选择要快捷得多。

错误信息不够人性化

Visual Studio 如果编译出错，错误信息只是简单的列在输出窗口里，要你去一个一个点开来看。而 Eclipse 的错误信息同时会在 Package Explorer 里显示成 Overlay Icon，从而清楚的了解到项目的哪些部分受到了错误的影响。

添加新类过于简单

Visual Studio 在创建新文件时唯一允许你指定的选项是文件名，剩下的只能手工修改。Eclipse 在创建新类时提供诸多选项，比如要求实现某个接口，那么所有接口方法的存根也会一并生成，节约了很多时间。

同一文件的导航功能不够一致

在 Visual Studio 的代码文件中如何导航？使用成员下拉框。

在可视化编辑器里怎么导航？用 Document Outline 窗口。

在引用的程序集里又如何导航？用 Object Browser 窗口。

在 Eclipse 中，上述所有内容都可以通过 Outlilne 窗口完成，操作也完全一致，不像 Visual Studio 那样每个窗口都有不同的界面和操作方法。

管理引用

我们创建项目的时候总有一些库是经常要用到的，比如数据项目引用 NHibernate，Silverlight 项目引用 Toolkit，等等。Visual Studio 引用哪些程序集只能由我们手工查找，而 Eclipse 提供了 User Library 的管理功能，能够让我们将常用的库引用一次性导进来，比 Visual Studio 的 Add References 不知方便几许。

重构功能比较初级

Visual Studio 支持代码重构，Eclipse 也支持，但是你比较一下两者的菜单就知道，它们对重构的支持程度完全不是同一个级别的。

代码提示不够清楚美观

Visual Studio 的代码提示同样只有单一的文字格式，大段的文字看下来，很难抓住重点。

Eclipse 的代码提示格式是富格式内容，能突出显示重点，可导航的部分会显示成超链接，同时还提供附加的功能按钮，比 Visual Studio 要细致体贴得多。

缺少本地历史记录功能

Eclipse 有一个非常强大的功能是将最近的编辑历史记录保存在本地，这样即使你没有使用版本数据库也能跟踪修订信息、比较版本、还原历史记录，对一些实验性的项目或者不需要签入的开源项目来说是非常有用的，并且你也可以用它来记忆内容，整理思路。Visual Studio 则完全没有这种功能。

缺少智能化的快速修复

Eclipse 不仅找出编译错误，还能够为许多种类型的错误提供修复选项。这个修复选项确实相当的智能，对很多常见的编译错误都能找到合理的解决方案，比如下面的错误它就能猜到 getMessge()实际上是 getMessage()拼错了，我只需要在这一项上按回车就万事 OK。

Visual Studio 的只能机械的显示一些错误信息。有时候错误提示也会附带一些关于如何修正错误的提示，但大多比较死板，需要程序员靠自己的经验来修复错误。

最后一个不得不说的地方是，Eclipse 是完全免安装的。

这意味着什么呢？如果我哪天重新安装了系统，那么打开 Eclipse 马上可以再次使用，以前设置的所有选项和更新内容全部立即可用。而 Visual Studio 就不得不重新安装、打补丁、设置各种选项，如果运气不好安装过程中突然出了什么错，那你所有过程重来一遍吧，一整天就这样没了。更让人讨厌的是，VS2010 安装过程中竟然要重启两次，就算你非重启不可的话，难道不能把所有内容准备好然后一次性解决吗！？