

Visual C# / 2005 Code Snippets

by Craig Murphy

With recent discussions in the DG newsgroups, it has become clear that there is an interest to know what articles, code, reviews, etc. work with Visual C# Express Edition (and Visual Basic.NET). With that in mind, hopefully the title of this article will go some way to providing a clue that Code Snippets work with both the Express Editions and the fully-fledged Professional edition and upwards.

Introduction

Code snippets are one of those features that developers either love or hate. Some developers don't like the thought of the IDE appearing to write code on their behalf.

If you're new to code snippets, the following series of screenshots (Figure 1, 2 and 3) should give you some indication of what I'm talking about.

Suppose we have to write a new class, let's call it Person. We could simply write the entire class declaration ourselves, typing in each character one by one, in the monotonous fashion that we're used to...we've all written hundreds if not thousands of class declarations, right? And we might make the odd typo on the way, etc.

Using Visual C# or Visual Studio 2005, code snippets remove that monotony and remove the potential for typos. Almost the instant you type in the letter 'c' from class, the IDE opens a drop-down menu offering the code snippet for a C# class, as shown in Figure 1.

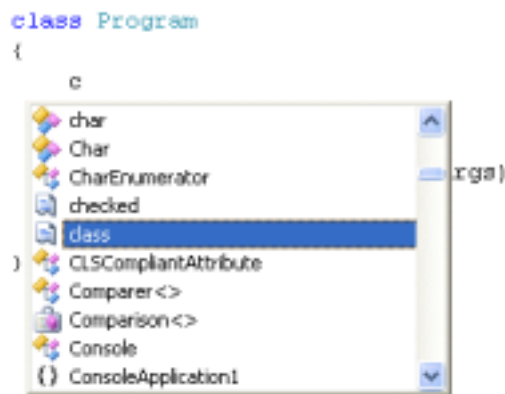


Figure 1

Pressing the tab key tells the IDE to accept the code snippet, resulting in Figure 2. You can see that it has finished off typing in the word 'class'. Pressing the tab a second time instructs the IDE to give us a class name and an opening and closing brace. OK, not so much time saved so far...and I would never call a class "MyClass". Luckily, the blue selection "MyClass" can be overwritten the second we start typing in the real name of our class.

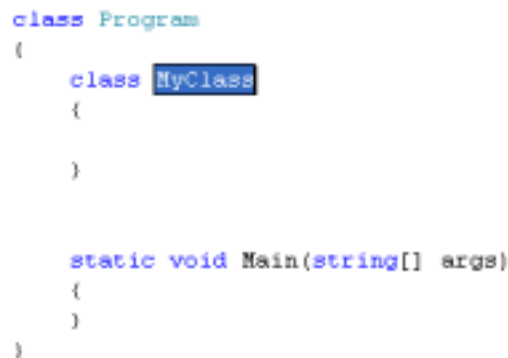


Figure 2

So we type in the word “Person”. As you are typing, the IDE replaces the blue selection with a green selection, as shown in Figure 3. What’s next? Normally we would move the cursor inside the curly braces. Well, pressing the Enter or Return key will do just that, and it will respect indentation too.

```
class Program
{
    class Person
    {
    }

    static void Main(string[] args)
    {
    }
}
```

Figure 3

Without splitting hairs, I counted 16 keystrokes for the manual version of the Person class versus 9 keystrokes for the version using code snippets. Throw in the occasional typo and I think code snippets suddenly become my new productivity enhancer.

Code Snippet Analysis

Whilst you may have thought that snippets are integrated into the IDE, and to some extent you would be right, they are actually defined in a text file with the .snippet extension. The snippet itself is represented using an XML format, and one that’s fairly easy to understand. Why are they represented using textual XML? Well, one reason is extensibility. You may wish to introduce custom snippets that are unique to the way you work.

There are plenty of snippets available for you to look at, by default there are some installed here:

C:\Program Files\Microsoft Visual Studio 8\VC#\Snippets\1033\Visual C#

Listing 1 presents the class.snippet XML.

```
<?xml version="1.0" encoding="utf-8" ?>
<CodeSnippets xmlns="http://schemas.microsoft.com/VisualStudio/2005/CodeSnippet">
  <CodeSnippet Format="1.0.0">
    <Header>
      <Title>class</Title>
      <Shortcut>class</Shortcut>
      <Description>Code snippet for class</Description>
      <Author>Microsoft Corporation</Author>
      <SnippetTypes>
        <SnippetType>Expansion</SnippetType>
        <SnippetType>SurroundsWith</SnippetType>
      </SnippetTypes>
    </Header>
    <Snippet>
      <Declarations>
        <Literal>
          <ID>name</ID>
          <ToolTip>Class name</ToolTip>
          <Default>MyClass</Default>
        </Literal>
      </Declarations>
      <Code Language="csharp"><![CDATA[class $name$
{
    $selected$$end$
}]]>
    </Code>
    </Snippet>
  </CodeSnippet>
</CodeSnippets>
```

Listing 1 – class.snippet

Our Own Example

It's all very well looking at the existing snippets: let's work through our own code snippet. Listing 2 presents a sample code snippet that lets us type in "ddgmb", expanding to "MessageBox.Show()", positioning the cursor inside the round brackets. The precise format of the code snippet file is well documented; there are links in the Links section at the end of this article. For the sake of this article, listing 2 has been saved on my Desktop as ddgmb.snippet.

```
<CodeSnippets xmlns="http://schemas.microsoft.com/VisualStudio/2005/CodeSnippet">

  <CodeSnippet Format="1.0.0">

    <Header>
      <Title>Developers Group Code Snippet</Title>
      <Shortcut>ddgmb</Shortcut>
      <Description>Code snippet for MessageBox.Show</Description>
      <Author>Craig Murphy</Author>
    </Header>

    <Snippet>
      <Code Language="VB">
        <![CDATA[MessageBox.Show($selected$end$)]]>
      </Code>
    </Snippet>

  </CodeSnippet>

</CodeSnippets>
```

Listing 2

Now, here comes the magic. We have to tell the IDE (Visual C# or Visual Studio) to import the code snippet file – it will take a copy of the snippet from wherever it is currently saved and will copy it to its own directory structure (typically C:\Documents and Settings\Craig\My Documents\Visual Studio 2005\Code Snippets\Visual C#\My Code Snippets – assuming that's where you choose to put the snippet, more about this in a moment). It is the Code Snippets Manager that "imports" code snippet in to the IDE for use. Figure 4 depicts the menus required to open the Code Snippet Manager.

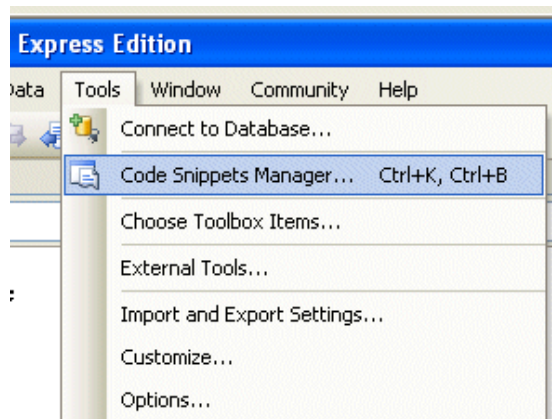


Figure 4

Figure 5 presents the Code Snippets Manager. As you can see, the Visual C# node is open and there's the class snippet that we saw earlier. Whilst it's not a major worry, I would like to see what the code snippet looks like in a preview window – it can be trial and error working out how a code snippet works. I suppose if I was that keen to understand a snippet, I could always load the XML .snippet file into Notepad and read through it.

The ddgmb code snippet can be used by creating a new Windows Application, dropping a button on the form, double clicking on the button and then pressing the letter 'd'. With any luck, Figure 8 should look familiar.

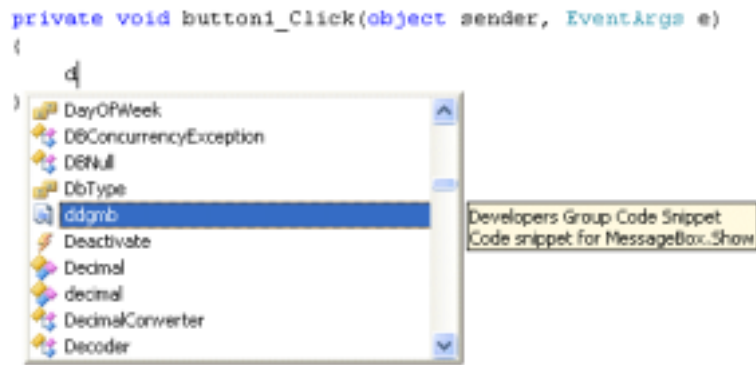


Figure 8

Pressing tab takes us to Figure 9, accepting the ddgmb code snippet.

```
private void button1_Click(object sender, EventArgs e)
{
    ddgmb|
}
```

Figure 9

Pressing tab again, takes us to Figure 10. Note the position of the cursor: it's ready for us to use.

```
private void button1_Click(object sender, EventArgs e)
{
    MessageBox.Show(|);
}
```

Figure 10

Life's Easier in Visual Basic

Whilst I've focused on using Visual C# for this article, code snippets also work in Visual Basic too. And it seems that folks on that side of the development world have things even easier. Listing 3 presents the same ddgmb.snippet file use in Visual Basic. Notice the addition of a <References> element. Its presence will instruct the IDE to add a reference to the specified assembly, should it be required. That's a useful time-saver, however it will only work for the Visual Basic language: we folks outside of the Visual Basic world will have to add our references manually.

```
<CodeSnippets xmlns="http://schemas.microsoft.com/VisualStudio/2005/CodeSnippet">
  <CodeSnippet Format="1.0.0">
    <Header>
      <Title>Developers Group Code Snippet</Title>
      <Shortcut>ddgmb</Shortcut>
      <Description>Code snippet for MessageBox.Show</Description>
      <Author>Craig Murphy</Author>
    </Header>
    <Snippet>
      <References>
        <Reference>
          <Assembly>System.Windows.Forms.dll</Assembly>
        </Reference>
      </References>
      <Code Language="VB">
```

```
<![CDATA[MessageBox.Show($selected$$end$)]]>
</Code>
</Snippet>

</CodeSnippet>

</CodeSnippets>
```

Listing 3

Conclusions

Code snippets are definitely a time saver and they should increase your productivity. Given that we can write our own code snippets, this should dispel the fear of the IDE writing code for us – we have total control over what code is automatically generated via the code snippet. Indeed, this control can be put to good use: perhaps with the creation of a corporate set of snippets that go some way to enforcing a particular coding style, etc.

I've not gone in to very much detail regarding the more complex usage scenarios for code snippets, I'll save that for a later article. In the meantime, I hope this brief look at code snippets and their use in the Express Editions of Visual Studio has whetted your appetite and the time savings that you make are put to good use.

Links

MSDN Code Snippets for Visual Studio 2005:

<http://msdn.microsoft.com/vstudio/downloads/codesnippets/>

Creating code snippets:

<http://msdn2.microsoft.com/en-us/library/ms165393.aspx>

Code snippet XML schema:

<http://msdn2.microsoft.com/en-us/library/ms171418.aspx>

Visual C# Express Edition – Free Forever:

<http://blogs.msdn.com/danielfe/archive/2006/04/19/579109.aspx>

Download Visual C# Express:

<http://msdn.microsoft.com/vstudio/express/visualcsharp/>

And the benefits of registering your copy of Visual C# Express:

<http://msdn.microsoft.com/vstudio/regbenefits/>



Craig is an author, developer, speaker, blogger, Certified ScrumMaster and Microsoft Most Valuable Professional (Connected Systems). He specialises in all things XML, particularly web services and XSLT. Craig is evangelical about .NET, C#, Test-Driven Development, Extreme Programming, agile methods and Scrum. He can be reached via e-mail at: ddg@craigmurphy.com, or via his web site: <http://www.craigmurphy.com> (where you can also find the source code and PowerPoint files for all of Craig's articles, reviews and presentations).