Visual Studio 2008 IDE Talk

Remember that there are many different keystrokes (hotkeys, keybindings) that can be used inside the IDE. You can find links to the various keybindings on my blog in this post:

<http://blogs.msdn.com/charlie/archive/2008/05/25/where-are-the-visual-studio-2008-keybinding-posters.aspx>

You might also find this post interesting:

<http://blogs.msdn.com/charlie/archive/2008/05/26/ide-color-schemes-for-the-vs-editor.aspx>

# Notes for the Talk: Coding

## Page 1

* Search –
  + Multiple Kinds of search
    - Find the right search for right job.
    - Incremental (Separate)
    - Quick search (Not Shown)
    - Find in Files
    - Find Symbol
  + **Tip #1 – Incremental search**
    - Search forward and back
    - Quick Search
      * Bookmark All
  + **Tip #2 – Find in Files**
    - Find and replace
    - Search across drives
    - Folder sets . Select files, type in new name, press apply
    - Two find results windows in Results Options
    - Replace in files
  + **Tip #3 – F3 repeats last search**
    - Shift F3 searches back.

## Page 2

* **Tip #4 – Box selection / reformat selection**
  + Use box selection to mess up a file
  + Use reformat to clean it up
    - Ctrl + K, D; Ctrl + K, F
    - Format selection might not do what you expect if you are below poorly formatted code.
* **Tip #5: Visible white space**
  + Edit | Advanced | View White Space
    - Ctrl + E, S
  + Set the color
  + Tools | Options | Text Editor | C# | Tabs | Keep Tabs
  + Tabify / Untabify Selected Lines
* **Tip #6: Guidelines**
  + Shut down shell
  + Look for reg file
    - [HKEY\_CURRENT\_USER\Software\Microsoft\VisualStudio\9.0\Text Editor]
    - RGB(255,0,0) 80
      * "Guides"="RGB(255,0,0) 80"
  + We should write a program to do this.
* **Tip #7: Clipboard ring**
  + Copy and cut two blocks of text
  + Paste them both
  + Ctrl-Shift-V
* **Tip #8: Toolbox General Tab**
  + Display Toolbox (Ctrl – W, X)
  + Drag text to General Tab
* **Tip #9: Tab Tab Snippit**
  + Run csc\_Snippit (See Source below)
* **Tip #10: Rename project**
  + Create a new console application
  + Add a constructor to your main class
  + Create an instance of the class in the main method
  + In the Solution Explorer, press F2 and rename the class
    - All names created above are changed.
  + Or create a new class, do the above, and instantiate it from main.
  + The rename will rename everything, even back in Main.

## Page 3

* **Tip #11: Command Window and Command line window**
  + Show how to run commands from the “search window” aka command line window
  + Ctrl + / to open command window
  + >alias fo File.OpenFile
  + >fo Program.cs
  + Ctrl+W,A to open the Command Window
  + Show that the same things work in that window.
* **Files Tab Channel**
  + Close all but this
  + Copy full path
  + Choose Copy Full Path. Open Notepad, Ctrl + O, paste, open file.
  + Show how to choose files in tabs with:
    - Ctrl + Alt + DownArrow
    - Ctrl + Tab
* **Tip #13: How to restore Ctrl + Tab**
  + Tools | Options
  + Window.NextDocumentWindow (Ctrl + F6)
  + Window.NextDocumentWindowNav (Ctrl + Tab)
  + Assign Ctrl+Tab to Window.NextDocumentWindow, if you want.
* **Tip #14: Navigate Forward and Backwards in Code**
  + Open SampleQueries project, turn to Program.cs
  + Use Ctrl+F (QuickFind) to search for harness 10 times
  + Use Ctrl+- to navigate back through searches
  + Forward: Ctrl+Shift + -
  + On Standard toolbar, point out Navigate Forward and Backward
  + Show dropdown on Navigate Forward.
  + IDE Navigator
  + Navigates to insertion points
  + Describe Ctrl + Shift + 7; and Ctrl + Shift + 8.
  + Bookmark all option in Quick Find

# Configure

* **Tip #1: Print Line Numbers**
  + Choose File | Print
  + Select “Include line numbers”
  + Name: Microsoft XPS Document Writer
  + Properties | XPS Documents
    - Check “Automatically Open XPS Documents”
  + Note Print Selection Option
  + Press Print and wait for document to show in window.
  + Default to best editor values
  + Tools | Options | Text Editor | C# | Line Numbers
* **Tip 2: Change size of fonts**
  + Tools | Options | Environment | Fonts and Colors
    - Show settings for | Environment Font
    - Choose a font, like Algerian, set it to 16.
  + In Editor: either
    - Assign Keystroke to
      * Macros.Samples.Accessibility.Decrease
      * Macros.Samples.Accessibility.Decrease
    - Bring Macro Explorer (Alt - F8)
      * Tools | Macros | Macro Explorer
      * Double click Increase and Decrease macros to change font size.
  + Show how to choose new template for fonts in editor
    - Tools | Import Export Settings
    - Import Selected Environment Settings
    - No, don’t save
    - Choose Ragnorak Blue
* **Tip #3: Make Statement Completion invisible**
  + Tools | Options | Environment | Fonts and Colors
  + Show Settings For: Statement Completion
    - Set the font size
* **Tip #4: Customize Toolbars**
  + Tools | Customize
  + Tabbed document
  + Hold down control to copy
  + Tools – Customize is now modal
    - You get to do more
* **Tip #5 – Full Screen Mode**
  + View | Full Screen
    - Toggle with: Shift + Alt + Enter
* **Tip #6 – 4 different window layouts**
  + Tools Import andExport Settings
    - Export selected environment settings.
    - Next, then unselect all
    - General Settings | Windows Layouts
    - Export them, then load them back in.
  + Windows menu
  + Reset Window layouts
  + Debug
  + Edit
  + Debug
  + Fullscreen
  + File (From command line only)
  + Current settings for layouts are stored in VSSettings
* **Tip #7: Shift + Esc**
  + Even Scott Guthrie didn’t know this one
  + Command going to tool windows?

## Page 2 of Customize

* **Tip #8 – Miscellaneous files project**
  + Tools | Options | Environment | Document
  + Show Miscellaneous Files in Solution Explorer
    - Set items saved to 5
  + File | Open | File (Ctrl + O}
  + Open a word document or some other document not part of the project
* **Tip #9 – Syntax Highlighting for any file**
  + Text syntax highlighting – set highlighting
* **Tip #10 – Create “throw away” projects**
  + Throw away projects that don’t get saved automatically
  + Tools | Options | Projects and Solutions | General
  + Save new projects when created
  + Zero impact projects (Zips)
* **Tip #11 – .NET Framework Multi-Targeting**
  + Target multiframework
  + MSBee – add on for msbuild for 1.1 framework

## Page 3 of Customize: Tools

* **Tip #12: VS Open to Last Project:** 
  + Open last project on startup
  + Environment | Startup – Load last loaded project
* **Tip #13 – Run external tools within IDE**
  + Tools | External Tools
  + Run the GUID Tool
  + Add a new tool
* **Tip #14 – Record macros**
  + Tools | macros

# Debugging

## Page 1: Breakpoints

* **Tip #1: Tracepoints** 
  + Right click on a line
  + Choose Breakpoints | New Tracepoint
  + Create a regular breakpoint
    - Right click, choose when Hit
* **Tip #2: Break at function**
  + Bring up Debug Window (Ctrl + D, B)
  + Choose New | Break at function (Ctrl + D, N)
* **Tip #3: Conditional breakpoint**
  + Right click on breakpoint
  + Simple LINQ code for looping over collection of integers
  + When hit count = 5
* **Tip #4: Disable breakpoints**
  + Ctrl + F9
  + In breakpoint window
    - Disable all breakpoints
    - Enable all breakpoints
    - Select items to enable disable

## Page 2: Examining Code

* **Tip #5: Step In, Over and Out**
* **Tip #6: Run to cursor**
  + Debug.RunToCursor
  + Ctrl + F10
* **Tip #7: Immediate Window**
  + Ctrl + D, I
  + Using simple LINQ code (Listing 1)
    - list.Count()
  + You can execute code at design time.
    - Create a method called ShowMe that returns a string and execute at design time
  + You can set a variable and use it:
    - int x = 2;
    - 2
    - x + 2
    - 4
    - x + 65
    - 67
* **Tip #8: Edit and continue** 
  + Using Simple LINQ code
    - Edit the list, adding 8 items
    - Move cursor back
    - Add Console.ReadLine();
    - Show results
* **Tip #9: Exception handling**

## Page 3: Debug Tips

* **Tip #10: Namespaces (From Peter Ritchie)**
  + Organize multi-project solution with namespaces
* **Tip #11: Using Statements**
  + Remove Unused Usings
  + Sort Usings
  + Remove and Sort Usings
* **Tip #12: Data Tips (Ahmed)**
  + Build Simple LINQ Database Application (Listing 2)
  + Investigate IQueryable by drilling in and show query
  + Use magnifying class to show Expression Tree
* **Tip #13: Class, Database diagrams (Joe Mayo)**
  + Using Simple LINQ Database Application
  + Drag database tables onto ORM Designer
* **Tip #14: Unload project (Before Build)** 
  + In Solution Explorer
  + Right click on project
  + Choose Unload Project
  + Choose Edit project file
  + See Listing 3 below for edited project that will copy a text file before build
  + http://pjbelfield.wordpress.com/2007/09/25/pre-build-events-and-the-before-build-target/
  + <http://blogs.msdn.com/manishagarwal/archive/2006/05/09/593510.aspx>

2) Start Visual Studio 2008 and bring up Tools > Options > Debugging > General.  If you are running under the **Visual Basic Profile**, you will need to check the box on the **lower left of the Options Dialog** marked "**Show All Settings**" before continuing (other profiles won't have this option).

Set the following two settings:

* Turn **OFF** the "**Enable Just My Code**" setting
* Turn **ON** the "**Enable Source Server Support**" setting

3) Next, bring up the "Symbols" Page and set the symbols download URL and a cache location.  Specifically, set the three settings below:

* + Set the symbol file location to be: **http://referencesource.microsoft.com/symbols**
  + Set a cache location.  Make sure this is a location that your account has read/write access to.  A good option for this is to place this path somewhere under your user hive (e.g. c:\users\sburke\symbols)
  + Enable the "Search the above locations only when symbols are loaded manually" option.

# Listings

**Listing 1: Simple Linq Code**

using System;

using System.Collections.Generic;

namespace ConsoleApplication58

{

class Program

{

static void Main(string[] args)

{

List<int> list = new List<int> { 1, 2, 3, 4, 5, 6, 7, 8 };

foreach (var item in list)

{

Console.WriteLine(item);

}

Console.ReadLine();

}

}

}

**Listing 2: Simple LINQ to SQL Code**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication59

{

class Program

{

static void Main(string[] args)

{

DataClasses1DataContext db = new DataClasses1DataContext();

var query = from c in db.Customers

where c.City == "London"

select c;

foreach (var item in query)

{

Console.WriteLine(item.CompanyName);

}

}

}

}

**Listing 3: Simple Build Step**

<Target Name="BeforeBuild">

       <Copy SourceFiles="c:\test.txt" DestinationFolder="d:\dest\"/>

</Target>

<Target Name="BeforeBuild">

<Copy SourceFiles="c:\users\ccalvert\documents\test.txt" DestinationFolder="c:\users\ccalvert\documents\test01"/>

</Target>

Listing 4: Test Snippet

<?xml version="1.0" encoding="utf-8" ?>

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

<CodeSnippet Format="1.0.0">

<Header>

<Title>csc\_Snippet</Title>

<Shortcut>csc\_Snippet</Shortcut>

<Description>Snippet for producing a snippet</Description>

<Author>Microsoft Corporation</Author>

<SnippetTypes>

<SnippetType>Expansion</SnippetType>

</SnippetTypes>

</Header>

<Snippet>

<Declarations>

<Literal Editable="true">

<ID>classname</ID>

<ToolTip>Class name</ToolTip>

<Default>csc\_TestSnippet</Default>

<Function>ClassName()</Function>

</Literal>

</Declarations>

<Code Language="csharp">

<![CDATA[<?xml version="1.0" encoding="utf-8" ?>

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

<CodeSnippet Format="1.0.0">

<Header>

<Title>$classname$</Title>

<Shortcut>$classname$</Shortcut>

<Description>$classname$</Description>

<Author>Microsoft Corporation</Author>

<SnippetTypes>

<SnippetType>Expansion</SnippetType>

</SnippetTypes>

</Header>

<Snippet>

<Declarations>

<Literal Editable="false">

<ID>classname</ID>

<ToolTip>Class name</ToolTip>

<Default>ClassNamePlaceholder</Default>

<Function>ClassName()</Function>

</Literal>

</Declarations>

<Code Language="csharp"><![CDATA[$end$]]

</Code>

</Snippet>

</CodeSnippet>

</CodeSnippets>]]>

</Code>

</Snippet>

</CodeSnippet>

</CodeSnippets>