

# Microsoft Visual Studio 2008

---

Default Keybindings  
Visual C#

Microsoft Student Partner  
Carles Gervilla Rivas

# Microsoft® Visual C#® Default Keybindings

This document contains the most commonly used keybindings that are available by default when selecting the Visual C# Development Settings. To reset the current development settings, go to Tools | Import and Export Settings, select **Reset All Settings**, and select **Visual C# Development Settings**.

To customize the keybindings for individual commands, go to Tools | Options | Environment | Keyboard. The keybindings in this document will appear under the Visual C# 2005 keyboard mapping scheme.

For a complete list of Visual C# default keybindings with their associated commands, see <http://go.microsoft.com/fwlink/?LinkID=100500>.

Notes: These keybindings are only available through the Visual C# Development Settings. To change to the Visual to the **Visual C# Development Settings**, go to Tools | Import and Export Settings and select **Reset All Settings** then select **Visual C# Development Settings**.

To customize the keybindings for these commands, go to Tools | Options | Environment | Keyboard.

## Editing

Toggle All outlining	<b>CTRL + M, L</b>	Toggles all previously collapsed outlining regions between collapsed and expanded states. <i>(Edit.ToggleAllOutlining)</i>
Toggle Outlining for Current Region	<b>CTRL + M, M</b>	Toggles the current region between collapsed and expanded states. <i>(Edit.ToggleOutliningExpansion)</i>
Collapse to Definitions	<b>CTRL + M, O</b>	Collapses existing regions to provide a high-level view of the types and members in the source file. <i>(Edit.CollapseToDefinitions)</i>
Stop Outlining	<b>CTRL + M, P</b>	Removes all outlining information from the whole document. <i>(Edit.StopOutlining)</i>
Comment and Uncomment	<b>CTRL + K, C</b> or <b>CTRL + E, C</b> and <b>CTRL + K, U</b> or <b>CTRL + E, U</b>	Inserts and removes, respectively, // at the beginning of the current line or every selected line. <i>(Edit.CommentSelection and Edit.UncommentSelection)</i>

Format Document	<b>CTRL + K, D</b> or <b>CTRL + E, D</b>	Formats the current document according to the indentation and code formatting settings specified on the <b>Formatting</b> pane under Tools   Options   Text Editor   C# <i>(Edit.FormatDocument)</i>
Format Selection	<b>CTRL + K, F</b> or <b>CTRL + E, F</b>	Formats the current selection according to the indentation and code formatting settings specified on the <b>Formatting</b> pane under Tools   Options   Text Editor   C# <i>(Edit.FormatSelection)</i>
Insert Snippet	<b>CTRL + K, X</b>	Displays the Code Snippet Picker. The selected code snippet will be inserted at the cursor position. <i>(Edit.InsertSnippet)</i>
Surround With	<b>CTRL + K, S</b>	Displays the Code Snippet Picker. The selected code snippet will be wrapped around the selected text.
Invoke Snippet From Shortcut	<b>TAB</b>	Inserts the expanded code snippet from the shortcut name. <i>(Edit.InvokeSnippetFromShortcut)</i>
Cycle Clipboard Ring	<b>CTRL + SHIFT + V</b>	Pastes text from the Clipboard ring to the cursor location in the file. Subsequent use of the shortcut key iterates through the item in the Clipboard ring.
Replace	<b>CTRL + H</b>	Displays the replace options in the <b>Quick</b> tab of the <b>Find and Replace</b> dialog box. <i>(Edit.Replace)</i>
Replace in Files	<b>CTRL + SHIFT + H</b>	Displays the replace option on the <b>In Files</b> tab of the <b>Find and Replace</b> dialog box. <i>(Edit.ReplaceInFiles)</i>
Show Smart Tag	<b>CTRL + PERIOD (.)</b> or <b>SHIFT + ALT + F10</b>	Displays the available options on the smart tag menu. <i>(View.ShowSmartTag)</i>

## File

New Project	<b>CTRL + SHIFT + N</b>	Displays the <b>New Project</b> dialog box. <i>(File.NewProject)</i>
Open Project	<b>CTRL + SHIFT + O</b>	Displays the <b>Open Project</b> dialog box, where existing projects can be added to the solution. <i>(File.OpenProject)</i>
Add Class	<b>SHIFT + ALT + C</b>	Displays the <b>Add New Item</b> dialog box and selects Class template as default. <i>(Project.AddClass)</i>

Add Existing Item	<b>SHIFT + ALT + A</b>	Displays the <b>Add Existing Item</b> dialog box, where existing files can be added to the current project. <i>(Project.AddExistingItem)</i>
Add New Item	<b>CTRL + SHIFT + A</b>	Displays the <b>Add New Item</b> dialog box, where a new file can be added to the current project. <i>(Project.AddNewItem)</i>
Show File List	<b>CTRL + ALT + DOWN ARROW</b>	Displays a pop-up listing of all open documents. <i>(Window.ShowEzMDIFileList)</i>
Open File	<b>CTRL + O</b>	Displays the <b>Open File</b> dialog box where a file can be selected to be opened. This does not add the file to the project. <i>(Edit.OpenFile)</i>

### IntelliSense

Complete Word	<b>CTRL + SPACE</b> or <b>CTRL + K, W</b>	Completes the current word in the completion list. <i>(Edit.CompleteWord)</i>
List Members	<b>CTRL + J</b> or <b>CTRL + K, L</b>	Invokes the IntelliSense® completion list. <i>(Edit.ListMembers)</i>
Quick Info	<b>CTRL + K, I</b>	Displays the complete declaration for the specified identifier in your code in a Quick Info tool tip. <i>(Edit.QuickInfo)</i>
Parameter Info	<b>CTRL + SHIFT + SPACE</b> or <b>CTRL + K, P</b>	Displays the name, number, and the type of parameters required for the specified method. <i>(Edit.ParameterInfo)</i>
Make Completion List Transparent	<b>CTRL</b>	Causes a visible completion list to become transparent.

### Navigation

Find All References	<b>SHIFT + F12</b> or <b>CTRL + K, R</b>	Displays a list of all references for the symbol selected. <i>(Edit.FindAllReferences)</i>
Go To Brace	<b>CTRL + ]</b>	Moves the cursor location to the matching brace in the source file. <i>(Edit.GoToBrace)</i>
Go To Definition	<b>F12</b>	Navigates to the declaration for the selected symbol in code. <i>(Edit.GoToDefinition)</i>
Go To	<b>CTRL + G</b>	Displays the <b>Go To Line</b> dialog box. <i>(Edit.Goto)</i>

Go To Next Location	<b>F8</b>	Moves the cursor to the next item, such as a task in the <b>Task List</b> window or a search match in the <b>Find Results</b> window. Subsequent invocations will move to the next item in the list. <i>(Edit.GoToNextLocation)</i>
Incremental Search	<b>CTRL + I</b>	Activates incremental search. If it is on, but no input is passed, the previous search query is used. If search input has been found, next invocation searches for the next occurrence of the input text. <i>(Edit.IncrementalSearch)</i>
Class View Go To Search, Combo	<b>CTRL + K, CTRL + V</b>	Brings focus to the <b>Class View search</b> box. <i>(View.ClassViewGoToSearch)</i>
Forward Browse Context	<b>CTRL + SHIFT + 7</b>	Moves to the next item called in code in the current file. Uses the Go To Definition navigation stack. <i>(View.ForwardBrowseContext)</i>
Pop Browse Context	<b>CTRL + SHIFT + 8</b>	Moves to the previous item called in code in the current file. Uses the Go To Definition navigation stack. <i>(View.PopBrowseContext)</i>
Navigate Backward	<b>CTRL + MINUS SIGN (-)</b>	Moves to the previously browsed line of code. <i>(View.NavigateBackward)</i>
Navigate Forward	<b>CTRL + SHIFT + MINUS SIGN (-)</b>	Moves to the next browsed line of code. <i>(View.NavigateForward)</i>
Find in Files	<b>CTRL + SHIFT + F</b>	Displays the <b>In Files</b> tab of the <b>Find and Replace</b> dialog box. <i>(Edit.FindInFiles)</i>
Find Symbol	<b>ALT + F12</b>	Displays the <b>Find Symbol</b> pane of the <b>Find and Replace</b> dialog box. <i>(Edit.FindSymbol)</i>
View Code	<b>F7</b>	Displays the selected item in <b>Code</b> view of the editor. <i>(View.ViewCode)</i>
View Designer	<b>SHIFT + F7</b>	Switches to <b>Design</b> view for the current document. Available only in <b>Source</b> view. <i>(View.ViewDesigner)</i>
View Markup	<b>SHIFT + F7</b>	Switches to <b>Source</b> view for the current document. Available only in <b>Design</b> view. <i>(View.ViewMarkup)</i>
Find	<b>CTRL + F</b>	Displays the <b>Quick</b> tab of the <b>Find and Replace</b> dialog box. <i>(Edit.Find)</i>

Move to Navigation Bar	<b>CTRL + F2</b>	Moves the cursor to the drop-down bar located at the top of the code editor when the editor is in <b>Code</b> view or <b>Server Code</b> view. <i>(Window.MoveToNavigationBar)</i>
Go to Find Combo	<b>CTRL + /</b>	Puts the cursor in the <b>Find / Command</b> box on the <b>Standard</b> toolbar. <i>(Edit.GoToFindCombo)</i>

## Build

Build Solution	<b>F6</b> or <b>CTRL + SHIFT + B</b>	Builds all the projects in the solution. <i>(Build.BuildSolution)</i>
Build Selection	<b>SHIFT + F6</b>	Builds the selected project and its dependencies. <i>(Build.BuildSelection)</i>

## Debugging

Autos	<b>CTRL + D, A</b>	Displays the <b>Autos</b> window, which displays variables used in the current line of code and the preceding line of code. <i>(Debug.Autos)</i>
Call Stack	<b>CTRL + D, C</b>	Displays the <b>Call Stack</b> window, which displays a list of all active methods or stack frames for the current thread of execution. <i>(Debug.CallStack)</i>
Immediate	<b>CTRL + D, I</b>	Displays the <b>Immediate</b> window, where expressions can be evaluated. <i>(Debug.Immediate)</i>
Locals	<b>CTRL + D, L</b>	Displays the <b>Locals</b> window, which displays the local variables and their values for each method in the current stack frame. <i>(Debug.Locals)</i>
Quick Watch	<b>CTRL + D, Q</b>	Displays the <b>QuickWatch</b> dialog box that has the current value of the selected expression. <i>(Debug.QuickWatch)</i>
Start	<b>F5</b>	Launches the application under the debugger based off of the settings from the startup project. When in Break mode, invoking this command will run the application until the next breakpoint. <i>(Debug.Start)</i>

Start Without Debugging	<b>CTRL + F5</b>	Launches the application without invoking the debugger. ( <i>Debug.StartWithoutDebugging</i> )
Step Into	<b>F11</b>	Executes code one statement at a time, following execution into method calls. ( <i>Debug.StepInto</i> )
Step Out	<b>SHIFT + F11</b>	Executes the remaining lines of a method in which the current execution point is located. ( <i>Debug.StepOut</i> )
Step Over	<b>F10</b>	Executes the next line of code, but does not follow execution through any method calls. ( <i>Debug.StepOver</i> )
Stop Debugging	<b>SHIFT + F5</b>	Stops running the current application under the debugger. ( <i>Debug.StopDebugging</i> )
Toggle Breakpoint	<b>F9</b>	Sets or removes a breakpoint at the current line. ( <i>Debug.ToggleBreakpoint</i> )
Watch	<b>CTRL + D, W</b>	Displays the <b>Watch</b> window, which displays the values of selected variables or watch expressions. ( <i>Debug.Watch</i> )
Enable Breakpoint	<b>CTRL + F9</b>	Toggles the breakpoint between disabled and enables. ( <i>Debug.EnableBreakpoint</i> )
Make Datatip Transparent	<b>[CTRL]</b>	Causes a visible datatip to become transparent.

## Window

Class View	<b>CTRL + W, C</b>	Displays the <b>Class view</b> window. ( <i>View.ClassView</i> )
Code Definition	<b>CTRL + W, D</b>	Displays the <b>Code Definition</b> window. ( <i>View.CodeDefinitionWindow</i> )
Command Window	<b>CTRL + W, A</b>	Displays the <b>Command</b> window, where commands can be invoked to manipulate the integrated development environment (IDE). ( <i>View.CommandWindow</i> )
Error List	<b>CTRL + W, E</b>	Displays the <b>Error List</b> window. ( <i>View.ErrorList</i> )
Object Browser	<b>CTRL + W, J</b>	Displays the <b>Object Browser</b> . ( <i>View.ObjectBrowser</i> )
Output	<b>CTRL + W, O</b>	Displays the <b>Output</b> window, where status messages can be viewed at run time. ( <i>View.Output</i> )

Properties Window	<b>CTRL + W, P</b>	Displays the <b>Properties</b> window, which lists the design-time properties and events for the currently selected item. <i>(View.PropertiesWindow)</i>
Solution Explorer	<b>CTRL + W, S</b>	Displays <b>Solution Explorer</b> , which lists the projects and files in the current solution. <i>(View.SolutionExplorer)</i>
Task List	<b>CTRL + W, T</b>	Displays the <b>Task List</b> window, which displays custom tasks, comments, shortcuts, warnings, and error messages. <i>(View.TaskList)</i>
Tool Box	<b>CTRL + W, X</b>	Displays the <b>Toolbox</b> , which contains controls that can be included or used with your code. <i>(View.Toolbox)</i>
Server Explorer	<b>CTRL + W, L</b>	Displays <b>Server Explorer</b> , which lets you view and manipulate database servers, event logs, message queues, Web services, and other operating system services. <i>(View.ServerExplorer)</i>
Close Tool Window	<b>SHIFT + ESC</b>	Closes the current tool window. <i>(Window.CloseToolWindow)</i>
Show Data Sources	<b>SHIFT + ALT + D</b>	Displays the <b>Data sources</b> window. <i>(Data.ShowDataSources)</i>
Close Document Window	<b>CTRL + F4</b>	Closes the current tab. <i>(Window.CloseDocument)</i>
Next Document	<b>CTRL + TAB</b>	Displays the IDE Navigator, with the first document window selected. <i>(Window.NextDocument)</i>

## Refactoring

Encapsulate Field	<b>CTRL + R, E</b>	Displays the <b>Encapsulate Field</b> dialog box, which allows creation of a property from an existing field and updates all references to use the new property. <i>(Refactor.EncapsulateField)</i>
Extract Interface	<b>CTRL + R, I</b>	Displays the <b>Extract Interface</b> dialog box, which allows creation of a new interface with members derived from an existing class, struct, or interface. <i>(Refactor.ExtractInterface)</i>
Extract Method	<b>CTRL + R, M</b>	Displays the <b>Extract Method</b> , dialog box, which allows creation of a new method from the selected code. <i>(Refactor.ExtractMethod)</i>

Promote Local Variable to Parameter	<b>CTRL + R, P</b>	Moves a variable from a local usage to a method, indexer, or constructor parameter and updates all call sites appropriately. <i>(Refactor.PromoteLocalVariableto-Parameter)</i>
Remove Parameters	<b>CTRL + R, V</b>	Displays the <b>Remove Parameters</b> dialog box, which allows removal of parameters from methods, indexers, or delegates by changing the declaration at any locations where the member is called. <i>(Refactor.RemoveParameters)</i>
Rename	<b>CTRL + R, R or F2</b>	Displays the <b>Rename</b> dialog box, which allows remaining all references for an identifier. <i>(Refactor.Rename)</i>
Reorder Parameters	<b>CTRL + R, O</b>	Displays the <b>Reorder Parameters</b> dialog box, which allows changes to the order of the parameters for methods, indexers and delegates.

© 2007 Microsoft Corporation. All rights reserved. Microsoft, IntelliSense, Visual C#, Visual Studio are trademarks of the Microsoft group companies. All other trademarks are property of their respective owners.