**Features of Visual Studio**

1. **Code and Text Editor**

The editor is the word processor of the integrated development environment (IDE). When it is used to edit text, it is referred to as the text editor. When it is used to edit source code in a Visual Studio development language, which is its more common use, it is referred to as the code editor.

A list of all documents open for editing in instances of the code editor is available on the **Windows** menu. The HTML Designer, CSS Editor, and XML Editor features of the IDE also employ this editor.

For most development languages, the code editor offers these features:

* Access to object properties, methods, and events at design time.
* IntelliSense statement completion.
* Collapsible code sections.
* A **Code Definition Window** that displays the source code for an object or element.
* A **Code Snippet Inserter** that lets you add ready-made blocks of model code.
* Options to define indents, tabs, and drag-and-drop behaviour.
* Unicode code pages.

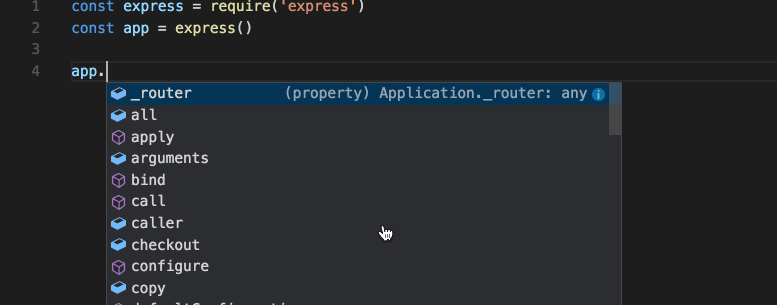
**User Interface Element List**

1. ***Code pane:*** The area where code or text is displayed for editing. It provides IntelliSense statement completion for the language in which you are developing. For more information, see [Using IntelliSense](https://msdn.microsoft.com/en-us/library/hcw1s69b(v=vs.100).aspx).
2. ***Indicator margin:*** A gray column on the left side of the code editor where indicators for breakpoints, bookmarks, and shortcuts are displayed. Clicking this area sets a breakpoint on the adjacent line of code.
3. ***Selection margin:*** A column between the indicator margin and the editing window where you can click to select lines of code. Changes to code are tracked here when you select the **Track Changes** option in the **Options** dialog box.
4. ***Horizontal and vertical scroll bars:*** Lets you scroll the code pane horizontally and vertically so that you can view the code that extends beyond the viewable edges of the pane.

**2. Intellisense**

IntelliSense is a general term for a variety of code editing features including: code completion, parameter info, quick info, and member lists. IntelliSense features are sometimes called by other names such as "code completion", "content assist", and "code hinting."

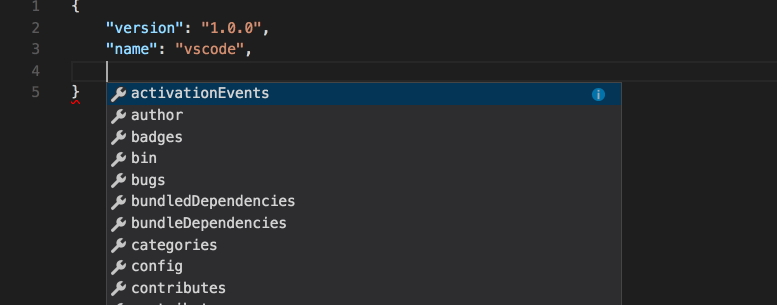
VS Code IntelliSense is provided for JavaScript, TypeScript, JSON, HTML, CSS, Less, and Sass out of the box. VS Code supports word based completions for any programming language but can also be configured to have richer IntelliSense by installing a language extension.



## IntelliSense features

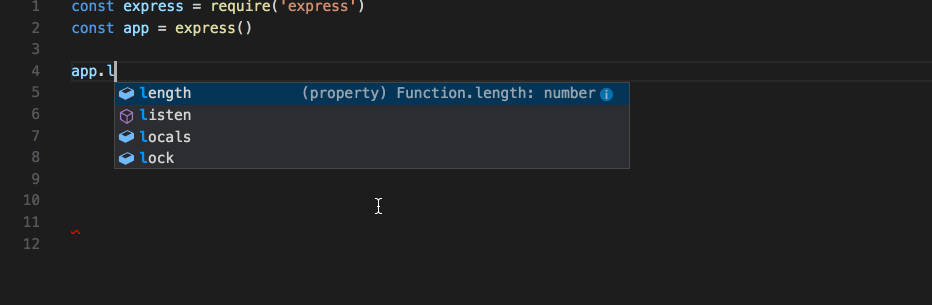
A language service provides intelligent code completions based on language semantics and an analysis of your source code. If a language service knows possible completions, the IntelliSense suggestions will pop up as you type. If you continue typing characters, the list of members (variables, methods, etc.) is filtered to include only members containing your typed characters. Pressing Tab or Enter will insert the selected member.

You can trigger IntelliSense in any editor window by typing Ctrl+Space or by typing a trigger character (such as the dot character (.) in JavaScript).



As provided by the language service, you can see **quick info** for each method by either pressing Ctrl+Space or clicking the info icon. The accompanying documentation for the method will now expand to the side.

The expanded documentation will stay so and will update as you navigate the list. You can close this by pressing Ctrl+Space again or by clicking on the close icon.



**3. Browser Link:**

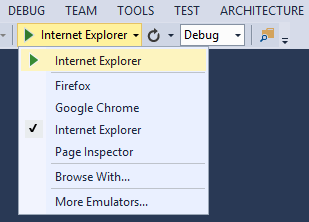
## a. Browser Refresh

With Browser Refresh, you can refresh multiple browsers that are connected to Visual Studio through Browser Link.

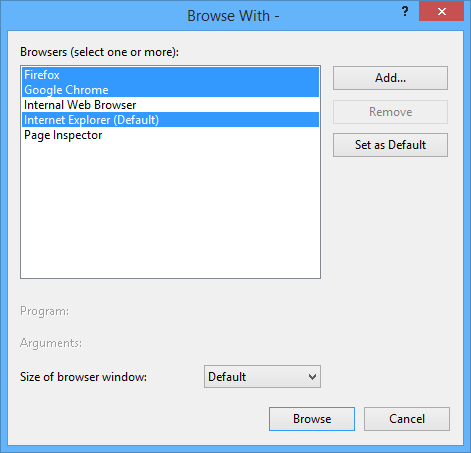
To use Browser Refresh, first create an ASP.NET application, using any of the project templates. Debug the application by pressing F5 or clicking the arrow icon in the toolbar:

C:\Users\Anupriya Kaushal\Desktop\image1.png

You can also use the dropdown to select a specific browser for debugging.



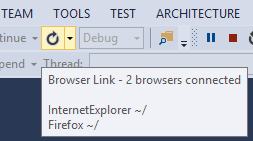
To debug with multiple browsers, select **Browse With**. In the **Browse With** dialog, hold down the CTRL key to select more than one browser. Click **Browse** to debug with the selected browsers. Browser Link also works if you launch a browser from outside Visual Studio and navigate to the application URL.



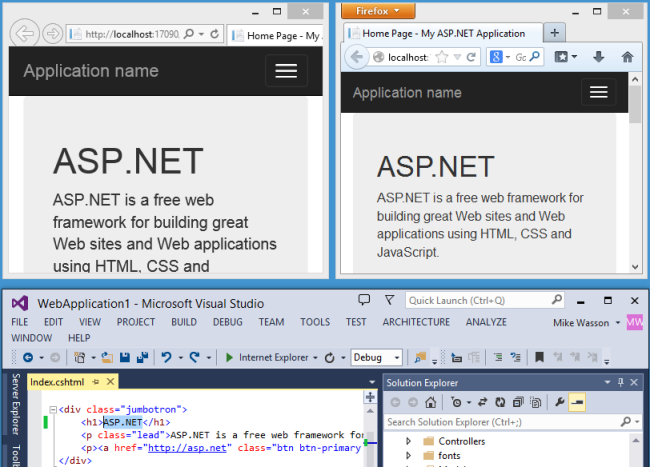
The Browser Link controls are located in the dropdown with the circular arrow icon. The arrow icon is the **Refresh** button.

C:\Users\Anupriya Kaushal\Desktop\image4.png

To see which browsers are connected, hover the mouse over the **Refresh** button while debugging. The connected browsers are shown in a ToolTip window.



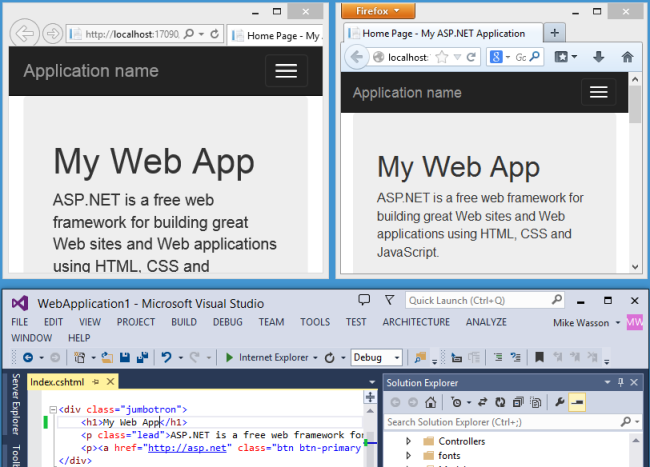
To refresh the connected browsers, click the **Refresh** button or press CTRL+ALT+ENTER. For example, the following screenshot shows an ASP.NET project, which I created using the MVC 5 project template. You can see the application running in two browsers at the top. At the bottom, the project is open in Visual Studio.



In Visual Studio, I changed the <h1> heading for the home page:



When I clicked the **Refresh** button, the change appeared in both browser windows:



## b. Enabling Browser Link for Static HTML Files

To enable Browser Link for static HTML files, add the following to your Web.config file.

XMLCopy

<configuration>

<system.webServer>

<handlers>

<add name="Browser Link for HTML" path="\*.html" verb="\*"

type="System.Web.StaticFileHandler, System.Web, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"

resourceType="File" preCondition="integratedMode" />

</handlers>

</system.webServer>

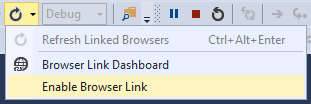
</configuration>

For performance reasons, remove this setting when you publish your project.

## c. Disabling Browser Link

Browser Link is enabled by default. There are several ways to disable it:

* In the Browser Link dropdown menu, uncheck **Enable Browser Link**.



* In the Web.config file, add a key named "vs:EnableBrowserLink" with the value "false" in the appSettings section.

<appSettings>

<add key="vs:EnableBrowserLink" value="false"/>

</appSettings>

* In the Web.config file, set debug to false.

<system.web>

<compilation debug="false" targetFramework="4.5" />

</system.web>

**4. Themes:**

Themes are made up of a set of elements: skins, cascading style sheets (CSS), images, and other resources. At a minimum, a theme will contain skins. Themes are defined in special directories in your Web site or on your Web server.

### Skins

A skin file has the file name extension .skin and contains property settings for individual controls such as [Button](https://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.button.aspx), [Label](https://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.label.aspx), [TextBox](https://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.textbox.aspx), or [Calendar](https://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.calendar.aspx) controls. Control skin settings are like the control markup itself, but contain only the properties you want to set as part of the theme. For example, the following is a control skin for a [Button](https://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.button.aspx) control:

<asp:button runat="server" BackColor="lightblue" ForeColor="black" />

### Cascading Style Sheets

A theme can also include a cascading style sheet (.css file). When you put a .css file in the theme folder, the style sheet is applied automatically as part of the theme. You define a style sheet using the file name extension .css in the theme folder.

### Page Themes

A page theme is a theme folder with control skins, style sheets, graphics files and other resources created as a subfolder of the \App\_Themes folder in your Web site. Each theme is a different subfolder of the \App\_Themes folder. The following example shows a typical page theme, defining two themes named BlueTheme and PinkTheme.

MyWebSite

App\_Themes

BlueTheme

Controls.skin

BlueTheme.css

PinkTheme

Controls.skin

PinkTheme.css

### Global Themes

A global theme is a theme that you can apply to all the Web sites on a server. Global themes allow you to define an overall look for your domain when you maintain multiple Web sites on the same server.

Global themes are like page themes in that they include property settings, style sheet settings, and graphics. However, global themes are stored in a folder named Themes that is global to the Web server. Any Web site on the server, and any page in any Web site, can reference a global theme.

**5. Debuggers:**

The Visual Studio debugger helps you observe the run-time behaviour of your program and find problems. The debugger works with all Visual Studio programming languages and their associated libraries. With the debugger, you can break execution of your program to examine your code, examine and edit variables, view registers, see the instructions created from your source code, and view the memory space used by your application.

Visual Studio includes a fully interactive source code debugger, providing a powerful and easy-to-use tool for tracking down bugs in your program. The debugger has complete support Visual Basic, C#, C/C++, and JavaScript.

The Visual Studio debugger is the common front end (that is, the user interface) to the debugging components that are, in turn, specific to the language being debugged. For new languages, all that is necessary for support by the Visual Studio debugger is to create the necessary back-end components, such as a debug engine (DE).