Visual Studio Tutorial

Visual Studio is a fully integrated development environment. It gives you a very sophisticated multiple-document-interface application in which you can do just about everything related to developing your code. This tutorial provides streamlined introductions which might be helpful to your development on our projects.

IDE Basics

- **Solution Explorer**
  The most important tool after creating a project in File Menu (new project or open project). With this tool, you can navigate through all files and items of the project.
  * If you cannot find specific windows, open the tools in View menu or right click the items in Solution Explorer.

- **Design View Window**
  If you are designing a UI application, you can use this window. This window presents a visual overview of what your form will look like. Also, the toolbox contains a large number of .NET components that you can drag onto your program.

- **Properties Window**
  You can see all properties of an item and configure it. (available when selecting items using the Design View)

- **Class View Window**
Show the hierarchy of the namespaces and classes in the project.

- **Object Browser Window**
  You can browse for and select existing component sets, and view the classes and members of the classes that are available within the subset.

- **Server Explorer Window**
  You can find information about services running, create new performance counts, and access the event logs. The *Data Connections* section enable not only connecting to existing databases and querying data, but creating a new database.

- **Integrated MSDN help**
  VS enable you to access the MSDN documentation from within the IDE by pressing the **F1 key** anywhere in the VS.

- **Quick Launch**
  It is located at top-right corner (or press Ctrl + Q). It is a fast way to find VS commands, tools, features, and so on.

---

### Debugging

1. **Setting Breakpoints**
   Debug > Windows > Breakpoints or click the line in the code editor with the shaded area.
2. Hover the mouse cursor over the name of the variable.
3. Click this magnifying class to use a debugger visualizer depending on the variable type.
4. Or, you can Add Watch for a variable which let you trace it including each values in an array in debug mode in the Locals window.

---

For detail figures and further instructions, you can take a look at this quick tutorial. [https://www.visualstudio.com/en-us/get-started/code/debug-your-app-vs](https://www.visualstudio.com/en-us/get-started/code/debug-your-app-vs)
**Version Control**

Git is supported in VS. We only need to setup some parameters through UI. The Team Explorer window is the main window including information of version control. The tutorial below has shown how to use built-in git commands.


Or, you can just use command-line interface to do the same thing.

**Backend Server**

When the application need to interact with the server, you can build a simple local server just for testing or remote access to an external server.

- **Local server**
  The most convenient way is to enable built-in Internet Information Services (IIS). IIS comes with Windows but is not turned on by default. Enabling IIS on the server creates a Web site server so that clients can use HTTP or HTTPS to connect.

![Windows Features](image)

For more setup details, please reference to the link below.

http://www.howtogeek.com/112455/how-to-install-iis-8-on-windows-8/

You will find a local web server whose folder is located at C:\inetpub\wwwroot, and can program (maybe with ASP.NET or JavaScript) to have some simple testing about communications between server.

- **Remote server**
  Server Explorer Window described above would let you access to remote database. With this tool, you can observe the change, query and create the remote database via VS.