Giới Thiệu Visual Studio

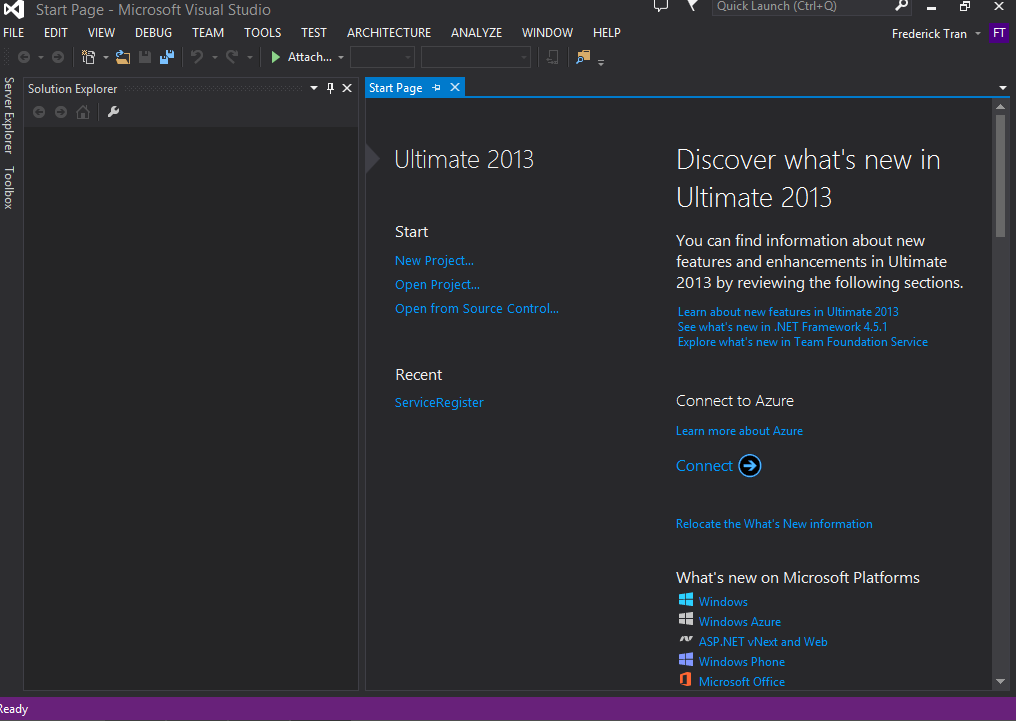
[**C/C++**](https://www.stdio.vn/articles/index/22/cplusplus)

Khái quát về Visual Studio

Microsoft Viual Studio (VS) là một IDE được Microsoft phát triển để hỗ trợ các lập trình viên trong quá trình viết mã. Ngoài chức năng cơ bản là viết mã, build và debug, VS còn cung cấp cho người dùng những chức năng như:

* Làm việc nhóm thông qua Team Foundation Server của Microsoft.
* Advanced Breakpoints.
* Các phím tắt và plugins hỗ trợ người dùng thao tác nhanh trong việc viết mã.
* Wizard Classes.
* Tùy chỉnh liên kết các project và thư viện, tập tin liên quan.
* ….

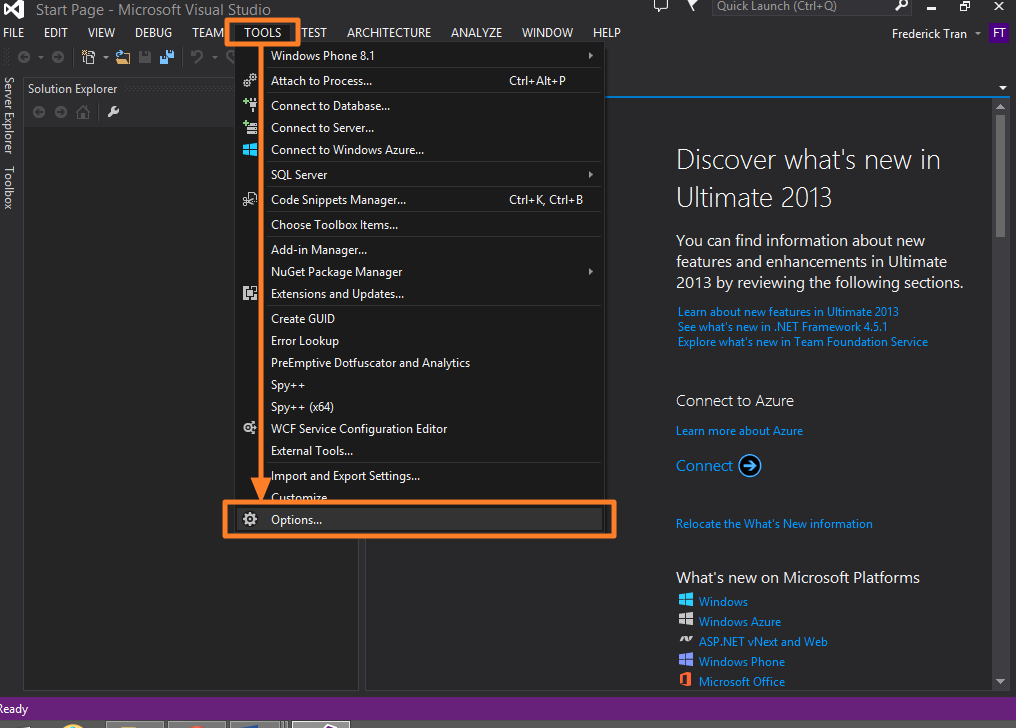
Đến thời điểm viết bài này thì phiên bản chính thức mới nhất của VS là 2013.



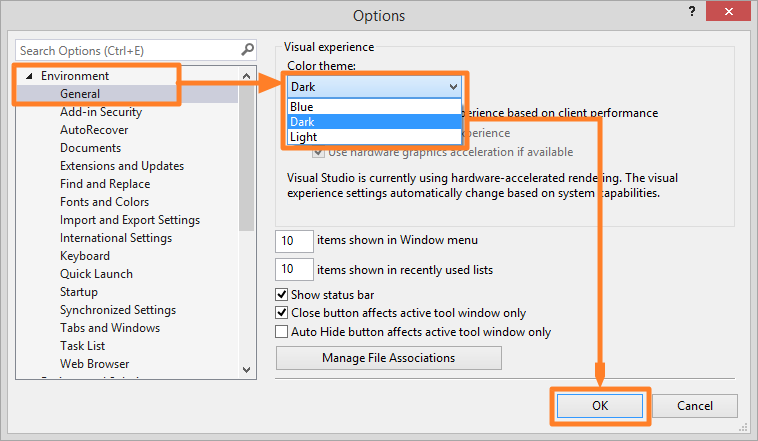
Visual Studio Color Themes

VS 2013 có 3 bộ theme chính phục vụ người dùng là Dark, Blue và Light. Điểm khác biệt của 3 bộ theme này chỉ là màu sắc của giao diện làm việc, các hình ảnh sử dụng trong bài viết thuộc về bộ theme Dark.

Để thay đổi màu sắc, người dùng chỉ cần vào **Tools -> Option**



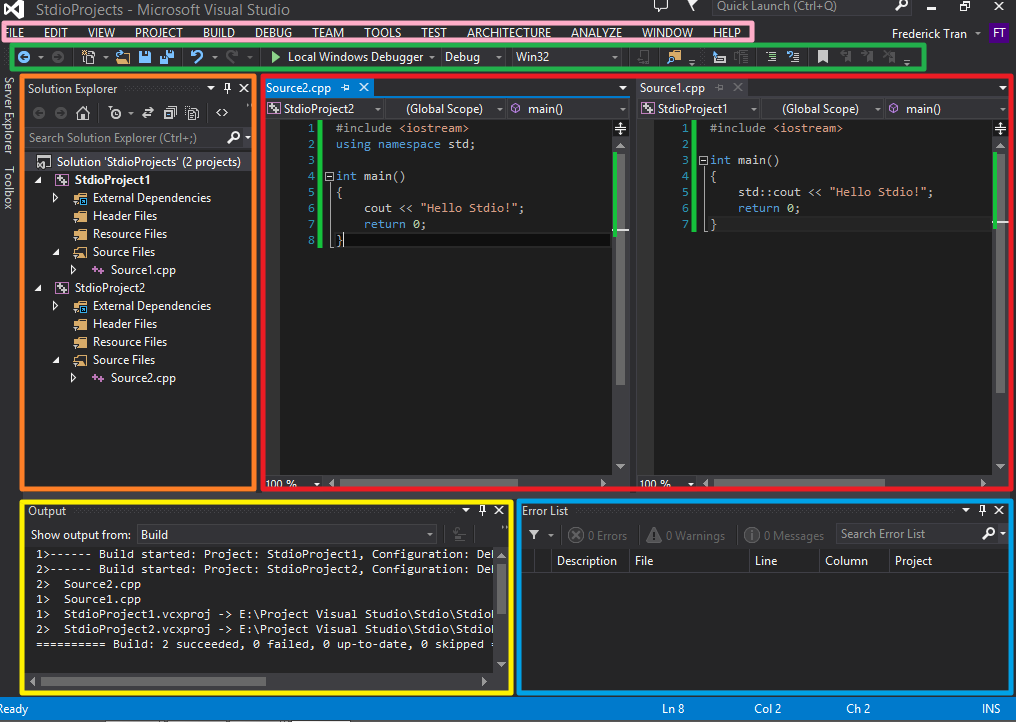
Tại cửa sổ vừa hiện lên, tại mục **General** chọn Color theme cần đổi và bấm **OK**.



Giao diện làm việc chính của Visual Studio

Để bắt đầu làm việc với VS, người dùng cần phải tạo hoặc mở ra một **Solution**và tạo hoặc mở **một hoặc nhiều Project** trong đó. Về cách tạo **Solution** và **Project**, tác giả [Rye Nguyễn](http://www.stdio.vn/users/index/74) đã có giới thiệu trong bài [Tạo Project C++ Đầu Tiên - Hello World](http://www.stdio.vn/articles/read/134/tao-project-c-dau-tien-hello-world).

Tôi đã chuẩn bị một Solution, và giao diện VS của tôi như sau:



**Solution Explorer**: là cửa sổ hiển thị Solution, các Project và các tập tin trong project.

**Khu vực code:** đây là khu vực để lập trình viên viết mã nguồn cho chương trình. Cửa sổ lập trình cho một tập tin trong Project sẽ hiển thị khi người dùng nháy đúp chuột lên tập tin đó trong cửa sổ **Solution Explorer**.

**Output**: đây là cửa sổ hiển thị các thông tin, trạng thái của Solution khi build hoặc của chương trình khi debug.

**Error List**: là cửa sổ hiển thị danh sách lỗi (Error) hoặc cảnh báo (Warning) của chương trình khi build.

**Toolbar** với các công cụ hỗ trợ người dùng trong việc viết mã và debug (các công cụ trên thanh có thể thay đổi khi bắt đầu debug).

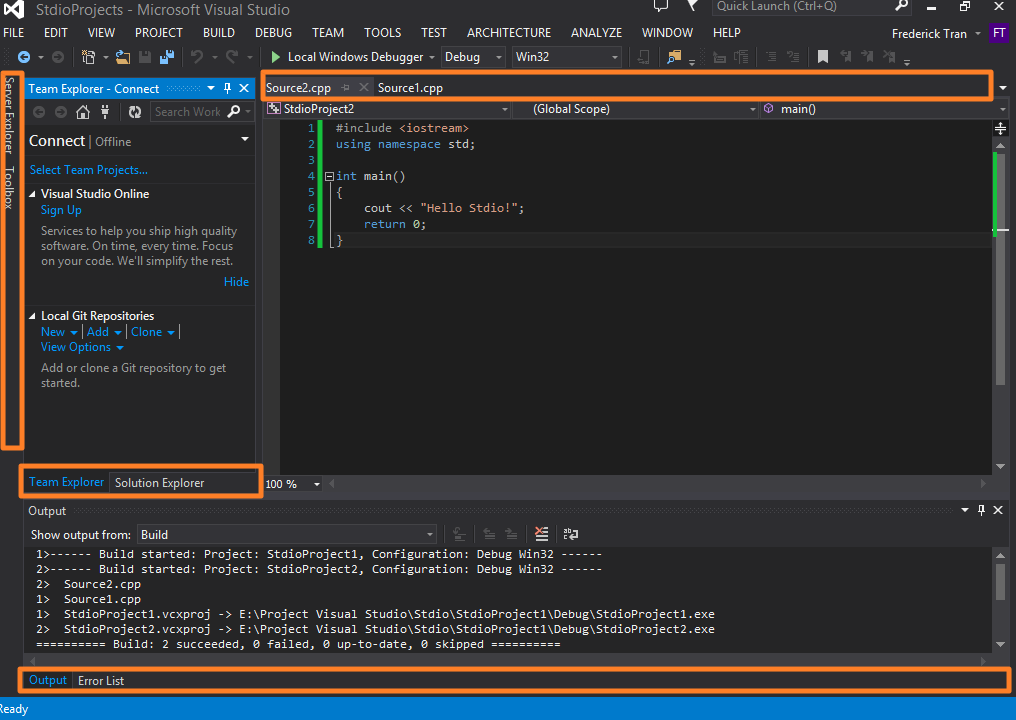
**Thanh menu** với đầy đủ các danh mục chứa các chức năng của VS. Khi người dùng cài thêm những trình cắm hỗ trợ VS (ví dụ như Visual Assist), thanh menu này sẽ cập nhật thêm menu của các trình cắm (nếu có).

Một lưu ý thêm là các cửa sổ trong giao diện làm việc của VS không được gắn cố định, nên người dùng có thể tự do tùy chỉnh khung làm việc của mình sao cho phù hợp nhất. Độc giả có thể thử click vào thanh tiêu đề của một cửa sổ hoặc một thẻ tập tin code và kéo ra khỏi vị trí của nó, một số biểu tượng hỗ trợ sẽ xuất hiện trên giao diện VS để người dùng “thả” cửa sổ vừa kéo ra vào đó hoặc thả trôi nổi như một popup trên giao diện VS.

Các cửa sổ bị ẩn trên giao diện Visual Studio

Đôi khi giao diện của VS trở nên quá dày đặc do quá nhiều cửa sổ, lúc này giải pháp được VS đưa ra là “gộp” nhiều cửa sổ lại với nhau và quản lý dưới dạng “thẻ”. Nội dung của cửa sổ sẽ được hiển thị khi ta click vào thẻ tương ứng.

Ví dụ về một số vị trí đặt thẻ trên giao diện VS:

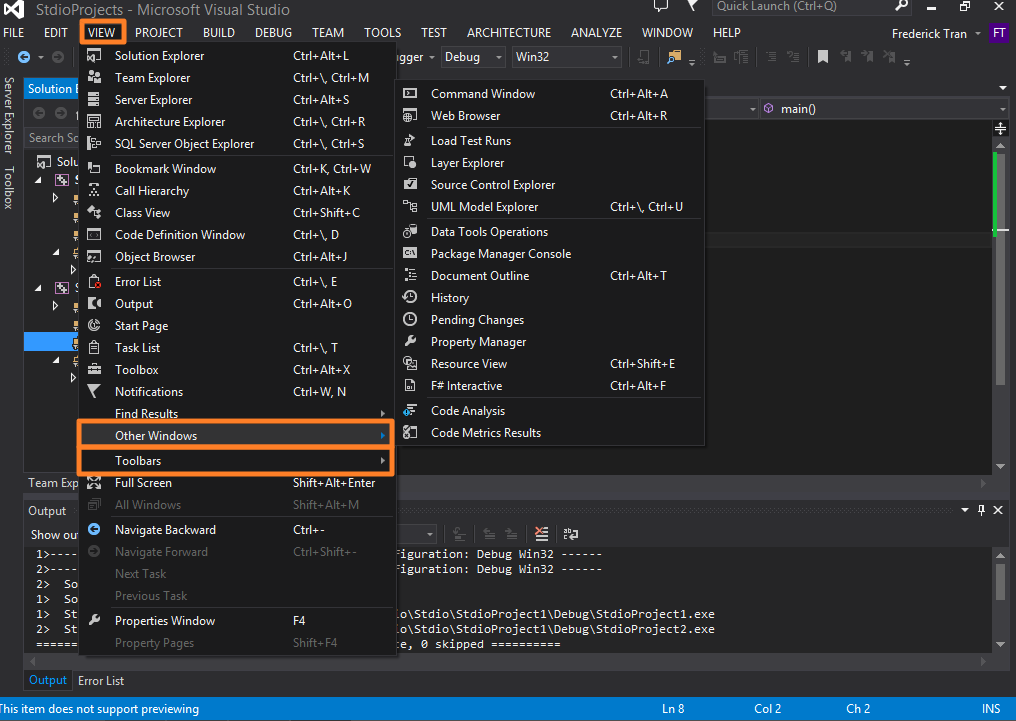


Thêm cửa sổ vào giao diện Visual Studio

Đôi khi một số cửa sổ không được kích hoạt sẵn trên giao diện cơ bản của VS, hoặc cũng có thể do người dùng vô tình tắt mất.

Để hiển thị cửa sổ mong muốn, người dùng có thể vào **View** và chọn cửa sổ cần hiển thị.

Một số cửa sổ được đặt trong **View -> Other Windows**, các thanh công cụ được đặt trong **View -> Toolbars**.



Build, Debug và Run

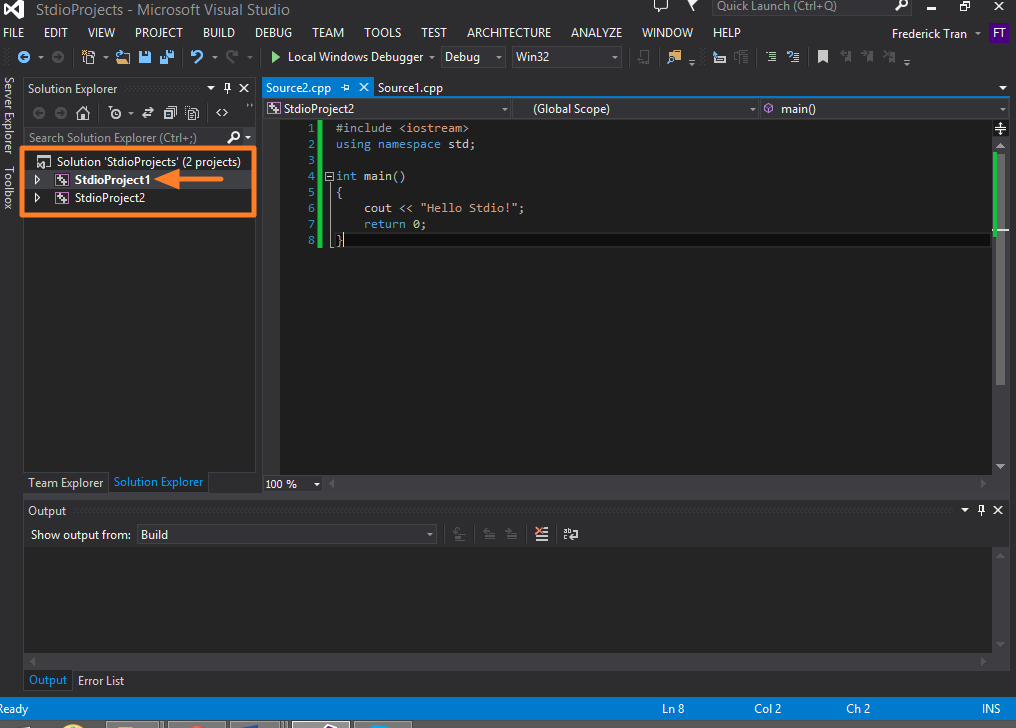
* Ở các phiên bản từ VS 2012 trở về trước, phím tắt mặc định để Build một Solution là **F7**, tuy nhiên ở phiên bản VS 2013, thao tác này được chuyển đến tổ hợp phím **Ctrl + Shift + B**.
* Để bắt đầu **Debug** một Solution, ta sử dụng phím tắt **F5**.
* Để **Run** một Solution, ta sử dụng tổ hợp phím **Ctrl + F5**.

\*Khác biệt giữa **Debug** và **Run**: Khi ta **Run** 1 Solution, tức là ta chạy trực tiếp chương trình sinh ra từ Solution đó trên Windows, những Breakpoint ta đặt trên cửa sổ code của VS sẽ không được bắt, và cửa sổ Output cũng sẽ không cập nhật trạng thái của chương trình đang chạy, nói cách khác là ta đang **“chạy thật”** chương trình đó trên máy. Với những ví dụ của tôi (mã nguồn từ ảnh), độc giả có thể thấy sự khác biệt trực quan hơn: khi ta **Run**, chương trình sẽ dừng lại cho ta xem kết quả, kèm theo một dòng ký tự “**Press any key to continue…!”** còn **Debug** thì không.

Xác định Project khởi động chương trình

Đối với những Solution có nhiều Project, vấn đề xảy ra khi ta sử dụng phím tắt để chạy chương trình. Giả sử mỗi Project là một chương trình riêng biệt, có hàm **main** riêng, làm sao để VS biết ta cần chạy hay debug project nào?

Nếu có nhiều Project, độc giả có thể thấy rằng: trong các project đó có một project được đặt tên in đậm – đó là **Startup Project**, khi ta khởi chạy hoặc debug từ phím tắt thì hàm **main** trong project này sẽ được gọi.



Để thiết lập **Startup Project**ta cần click phải chuột lên project muốn đặt làm startup và chọn **Set as Startup Project**.

