

**Công ty Cổ phần Chứng khoán FPT**

Tài liệu Đào tạo Windows Services

**Hà Nội, <30-03-2018>**

Record of change

\*A - Added M - Modified D - Deleted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ngày | Hạng mục thay đổi | A\* M, D | Mô tả thay đổi | Version |
| 27-02-2020 |  | A | Tạo tài liệu | 1.0 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

trang ký

**NGƯỜI LẬP:** <Name> <Date>

<Position>

**NGƯỜI XEM XÉT:**

<Name> <Date>

<Position>

<Name, if it’s needed> <Date>

<Position>

**NGƯỜI PHÊ DUYỆT:**

<Name> <Date>

<Position>

MỤC LỤC

I. Tổng quan 5

II. Tạo ứng dụng loại Windows Service. 6

Bước 1: 6

Bước 2: 6

Bước 3: 6

III. Thêm vào thành phần Installer 7

Bước 4: 7

Bước 5: 8

Bước 6: 8

Bước 7: 9

IV. Tạo một trình ứng dụng loại Setup để cài đặt (Install)Service: 9

Bước 8: 9

Bước 9: 11

Bước 10: 11

Bước 11 : 12

Bước 12 : 12

Bước 13 : 13

V. Cài đặt (Installing) và kiểm thử (test) Service 14

Bước 14 : 14

Bước 15 : 15

Bước 16 : 16

VI. Gỡ cài đặt (Uninstalling) 16

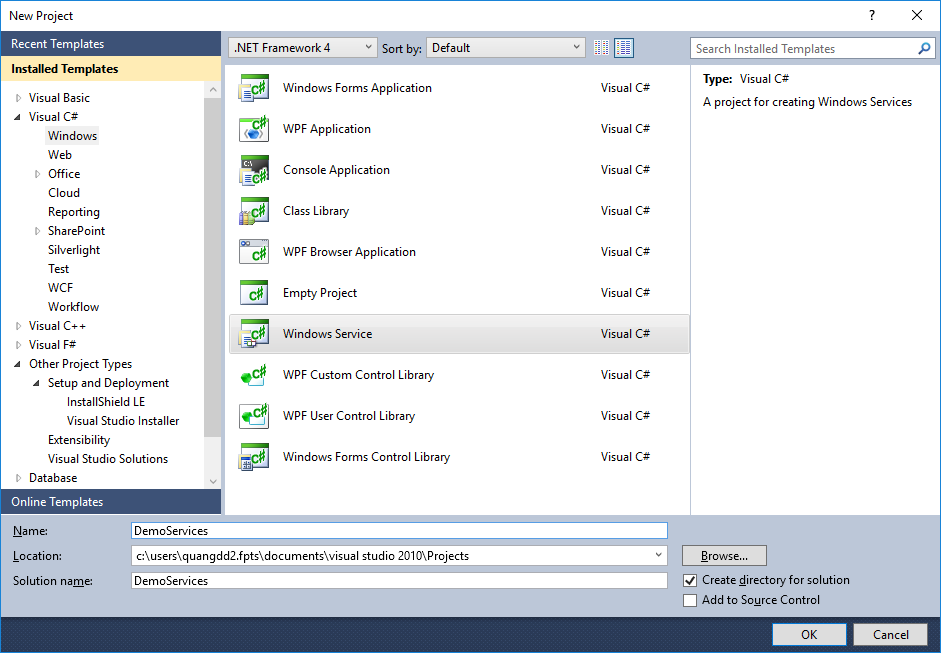
# Tổng quan

* **Windows Services** cung cấp phương tiện cho application logic chạy liên tục trên máy tính của bạn, thông thường là việc cung cấp điều khiển thiết bị hoặc các dịch vụ hệ điều hành.**Windows Services** là hữu dụng đối với các ứng dụng phía server mà ta muốn chúng phải luôn ở trạng thái sẵn sàng phục vụ các yêu cầu từ client. Nói chung, nếu bạn có nhu cầu xây dựng một trình ứng dụng mà bản chất hoạt động của nó mang tính “âm thầm làm việc” và không cần sự tương tác trực tiếp của người dùng, bạn có thể nghĩ ngay đến việc tạo trình ứng dụng loại **Windows Service**.
* **Windows Service** là một ứng dụng chạy trên máy server hoặc workstation và cung cấp những chức năng mà sự diễn tiến của nó không cần sự tương tác trực tiếp của người dùng. **Windows Services** thường được dùng để giám sát hoạt động hệ thống.
* Một **Windows Service** sẽ chạy trong tiến trình của riêng nó, không phụ thuộc người dùng hay các chương trình khác đang chạy trên cùng máy tính. **Windows Services** thường được cấu hình để tự động bắt đầu khi nào máy tính khởi động. Không giống như đa phần các ứng dụng, **Windows Services** chạy dưới nhận dạng bảo mật của chính nó, thay vì dưới nhận dạng của đăng nhập người dùng hiện hành. Chúng có thể bắt đầu chạy, ngay cả nếu không có người dùng (user) đăng nhập máy tính.
* Khi mà bạn tạo một ứng dụng với mục tiêu ứng dụng này sẽ chạy như là một **Windows Service**, bạn cần phải lưu ý rằng nó không bao gồm bất kỳ các yếu tố giao tiếp người dùng như **Message Boxes** hay **Dialog Boxes**. Một **Windows Service** không có tính chất cung cấp một giao diện trực quan (visual interface) cho người dùng.
* Thông thường một **Windows Service** sẽ báo cáo đặc thù các kết quả và thông báo lỗi của nó tới một sổ ghi biến cố (event log). Tuy nhiên bạn cũng có thể kết hợp nó với **Web Service** để làm những việc “động trời” hơn như là âm thầm chuyển thông tin từ máy mà nó đang được cài đặt đến một máy ở xa thông qua đường truyền internet.
* Nếu như bạn đã quen thuộc với**Microsoft SQL Server 2000**, bạn sẽ nhận thấy nó chạy như là một **Windows Service**. Một ví dụ dễ hiểu của ứng dụng **Windows Service** là **Windows Time Service**, nó cập nhật thông tin đồng hồ mà bạn thấy trên thanh taskbar của máy tính. Trước đây, việc phát triển các ứng dụng dạng này là rất khó khăn. Giờ đây, nhờ có **.NET Framework** chứa một tập các class, cung cấp các chức năng cơ bản cho việc cài đặt và phát triển dễ dàng các ứng dụng **Windows Service** trên nền **C# .NET**

# Tạo ứng dụng loại Windows Service.

## Bước 1:

Khởi động Visual Studio .NET và tạo một project mới với việc sử dụng mẫu Windows Service. Đặt tên cho project là DemoServices và chứa nó vào một thư mục trên máy tính của bạn.



## Bước 2:

Trên cửa sổ Solution Explorer đổi tên của component có tên Service1.cs thành MyServices.cs.

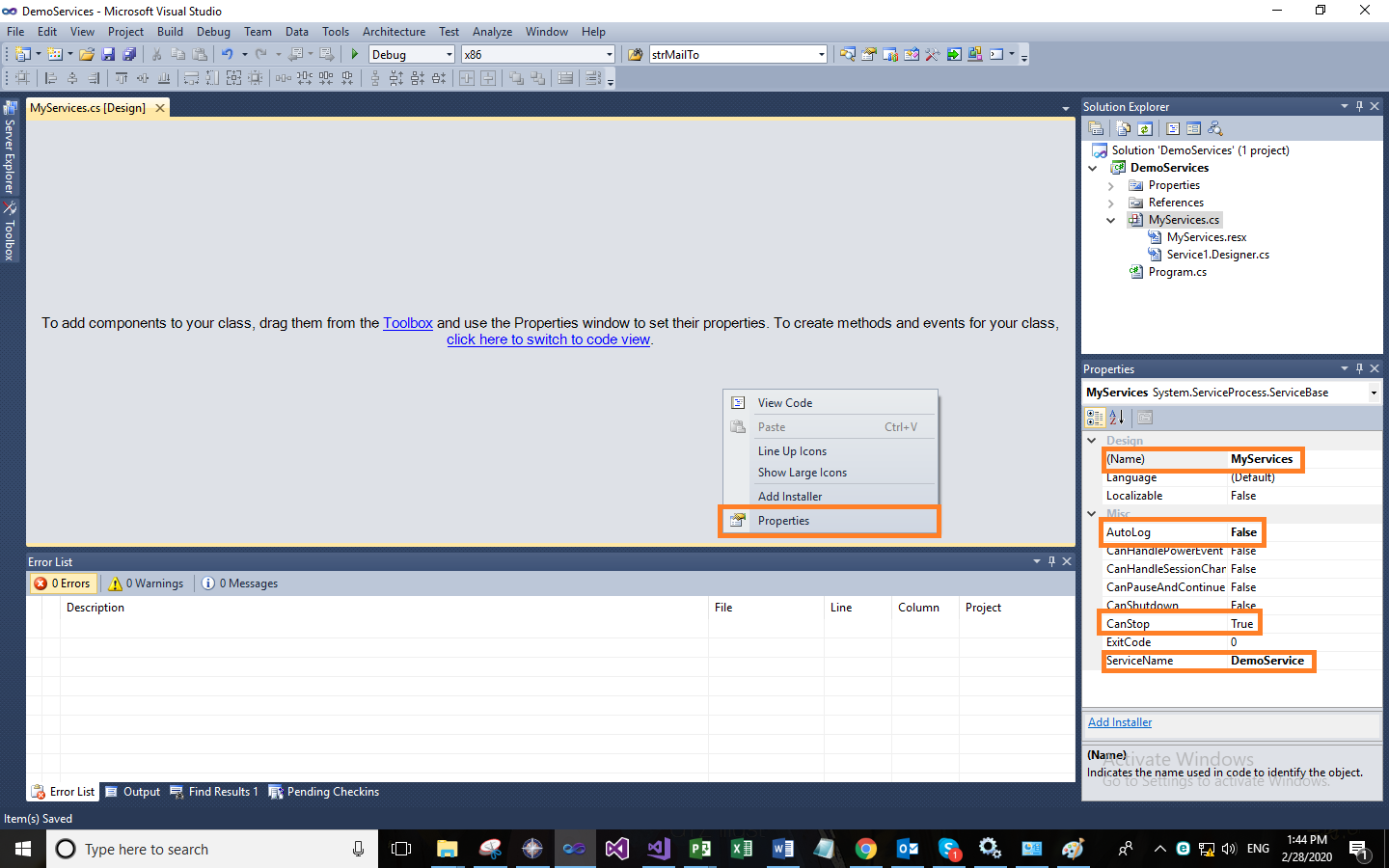
## Bước 3:

Trong cửa số **Solution Explorer**, Click chọn **MyServices.cs**. Trên cửa sổ **MyServices.cs[Design]** Click chuột phải chọn **Properties**. Trên cửa sổ Properties lần lượt thực hiện các công việc sau:

· Tại mục **ServiceName**: đổi tên thành **MyServices.cs**.

· Tại mục **AutoLog** : Thiết lập giá trị là **False**.

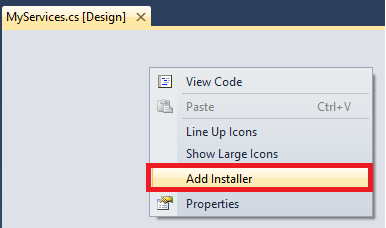
· Tại mục **CanStop**: Thiết lập giá trị là **True**.



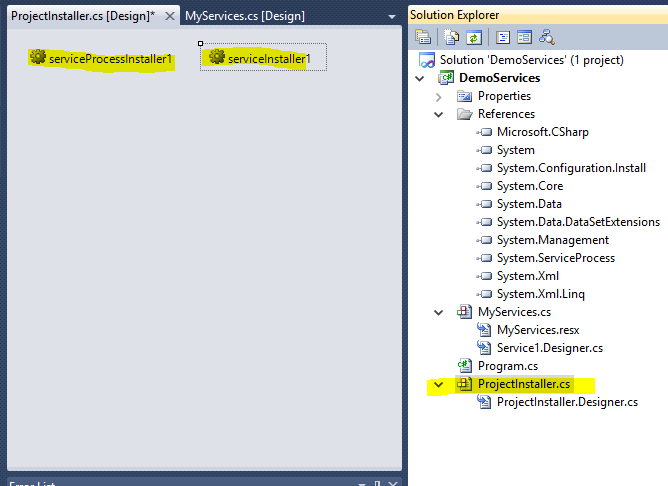
# Thêm vào thành phần Installer

## Bước 4:

Trên cửa sổ **MyServices.cs[Design]** Click chuột phải chọn **Add Installer** .

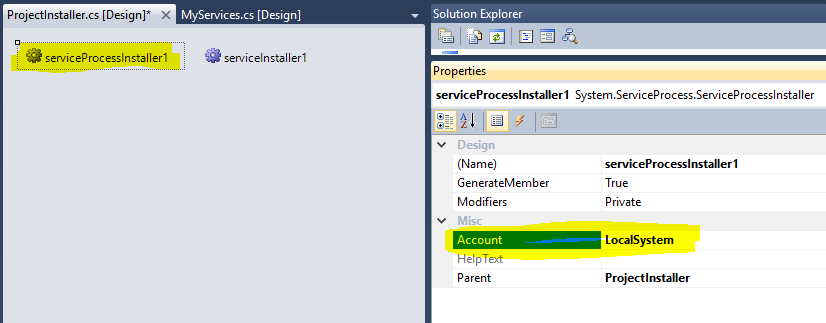


Một class có tên là*ProjectInstaller.cs* sẽ được thêm vào project của bạn. Bạn sẽ thấy ở màn hình design của thành phần này có hai biểu tượng thành tố khác trên nó là :*ServiceProcessInstaller1* và *ServiceInstaller1*.



## Bước 5:

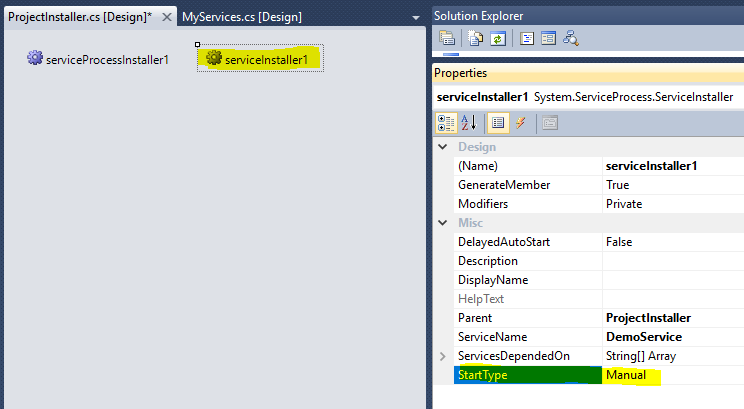
Click chuột phải vào *ServiceProcessInstaller1* và hiển thị cửa sổ **Properties** --> tại thuộc tính **Account**click chọn **LocalSystem** trong danh sách



## Bước 6:

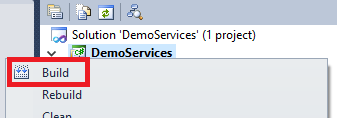
Click chuột phải vào *ServiceInstaller1* và hiển thị cửa sổ **Properties**. Chọn giá trị phù hợp cho thuộc tính **StartType:**

-**Manual:** Bạn sẽ phải bật/tắt thủ công khi muốn khởi động hoặc tắt Services

**-Automatic:** Tự động bật khi khởi động máy tính

## Bước 7:

Thực hiện việc build trình ứng dụng

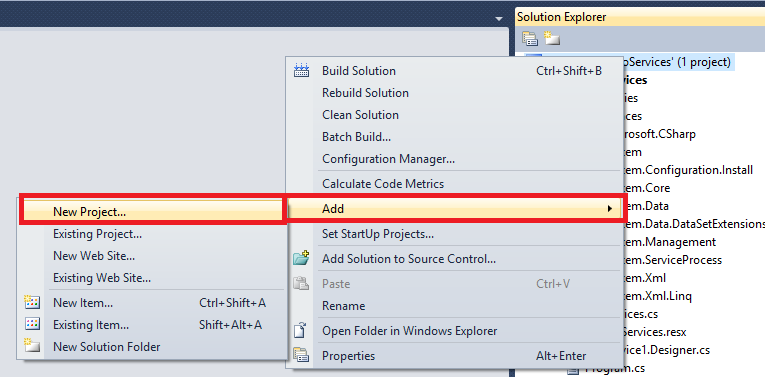


# Tạo một trình ứng dụng loại Setup để cài đặt (Install)Service:

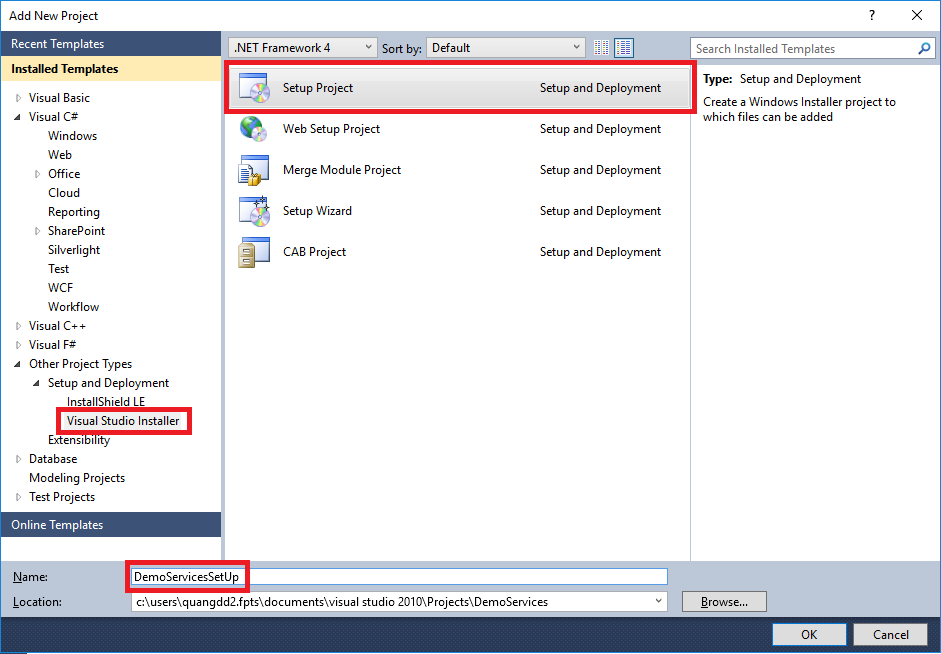
## Bước 8:

Trên thanh công cụ chọn **File** --> **Add Project** --> **New Project.**

Hoặc bạn có thể Click chuột phải vào **Solution🡪Add🡪 New Project.**



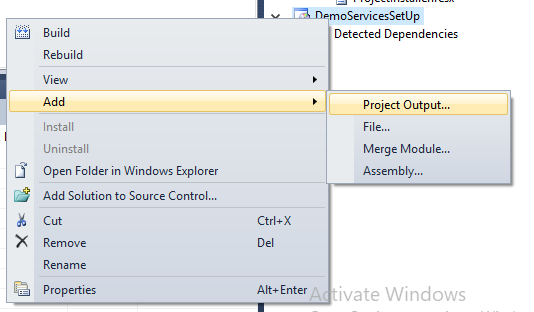
Đặt tên Projects là **DemoServicesSetup.  
Lưu ý: Nếu phiên bản Visual Studio bạn dùng không có tùy chọn Project như hình bên dưới thì bạn cần cài đặt thêm Package phù hợp.**



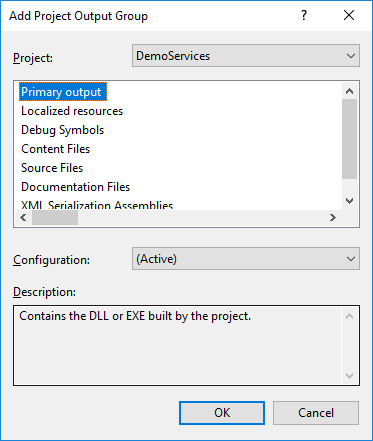
## Bước 9:

Trong cửa sổ **Solution Explorer**,

click phải chuột tại **DemoServicesSetup🡪 Add** 🡪**Project Output**



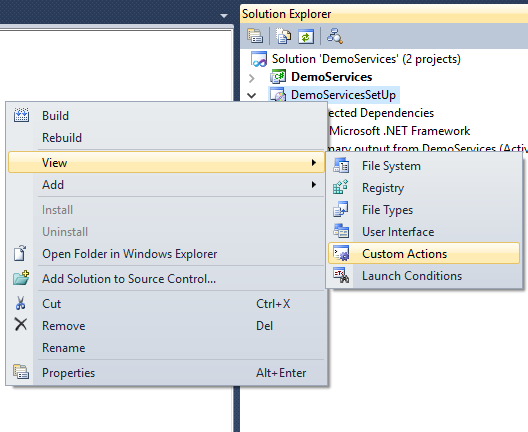
    Từ cửa sổ **Add Project Output Group** --> chọn **Primary Output** --> click OK



## Bước 10:

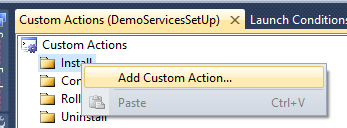
Trong cửa sổ **Solution Explorer**,

click phải chuột tại mục **DemoServicesSetup** 🡪**View** 🡪 **Custom Actions**



## Bước 11 :

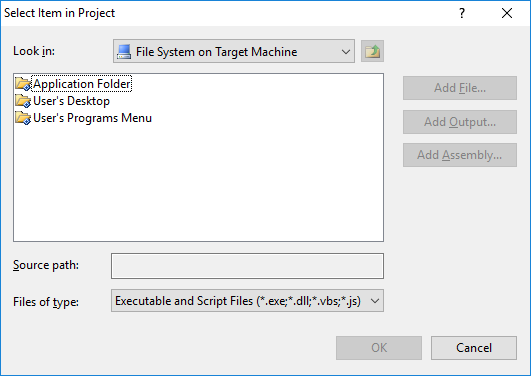
Trên cửa sổ **Custom Actions** --> click phải chuột vào **Install**--> **Add Custom Action**

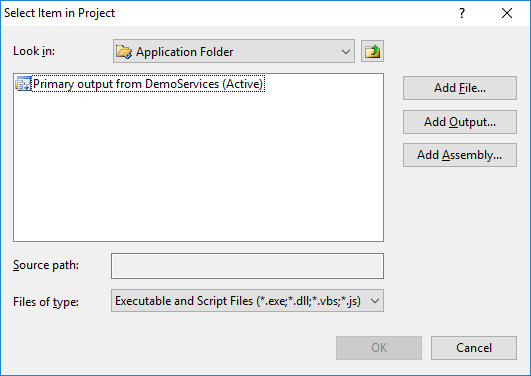


Thực hiện **Bước 17** tương tự với Mục **Uninstall**

## Bước 12 :

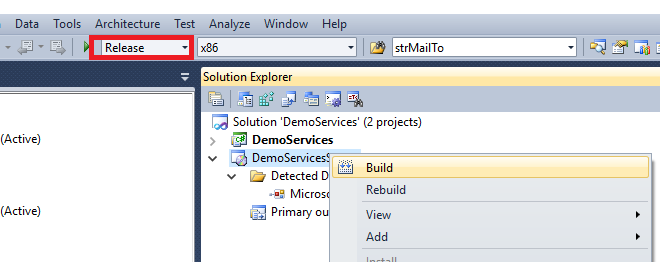
Xuất hiện cửa sổ **Select Item in Project** --> Double-click mục **Application Folder**, chọn mục *Primary Output from swGiamSatThuMuc (Active)* --> click OK.





## Bước 13 :

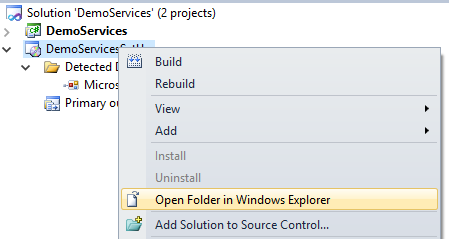
Thực hiện việc build trình ứng dụng **DemoServicesSetup**(chú ý nhớ chọn định cấu hình cho **Solution** là *Release* ).



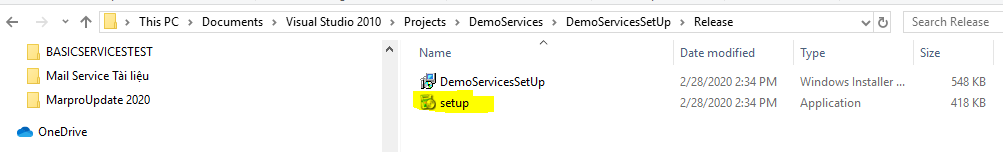
# Cài đặt (Installing)

## Bước 14 :

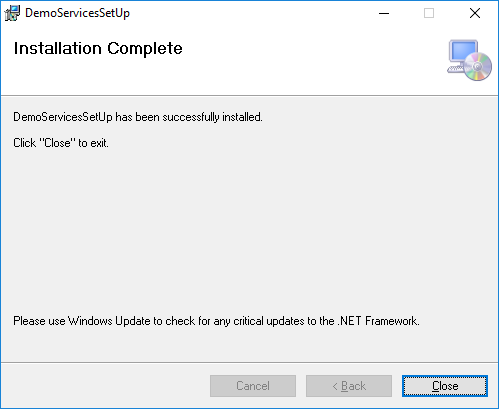
Click chuột phải vào **DemoServicesSetup**🡪**Open Folder in Windows Explorer**



Trong thư mục của project vừa tạo **DemoServicesSetup** --> bạn vào thư mục con có tên **Release** và tìm tập tin có tên **Setup**. Double-click tập tin này để start việc cài đặt (installation).

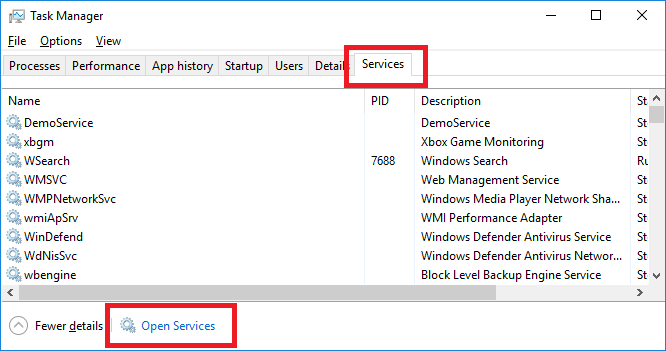


Thao tác này sẽ start trình **Setup Wizard**. Chấp nhận tất cả các giá trị ngầm định và hoàn tất việc cài đặt (installation).



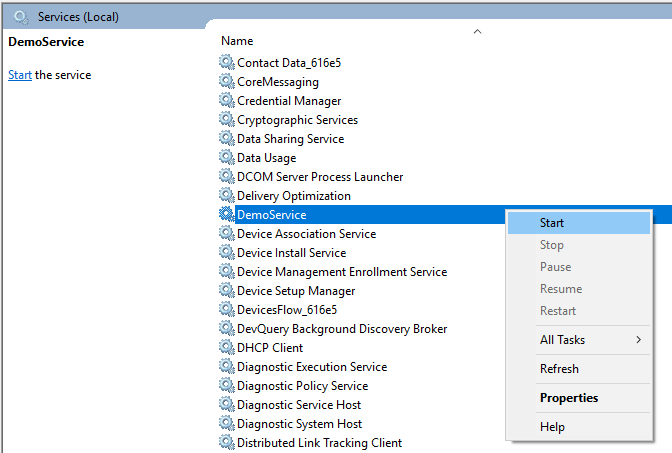
## Bước 15 :

Để kiểm tra service vừa được cài đặt, Mở **Task Manager**🡪 **Tab Service** 🡪 **Open Services**



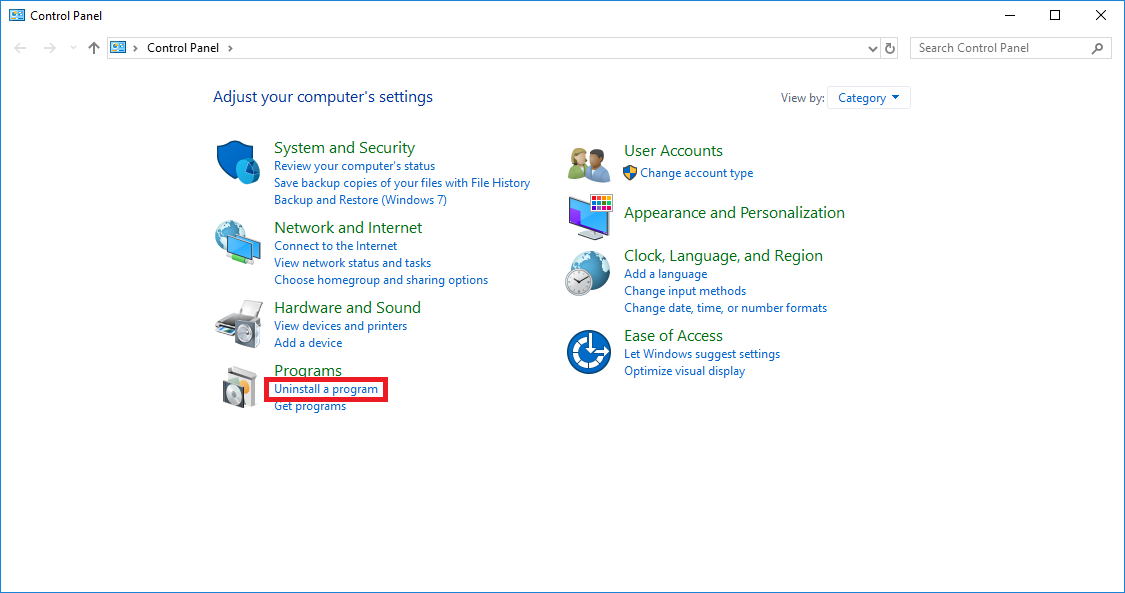
## Bước 16 :

Click phải trên service của bạn và chọn **Properties** --> **Start**

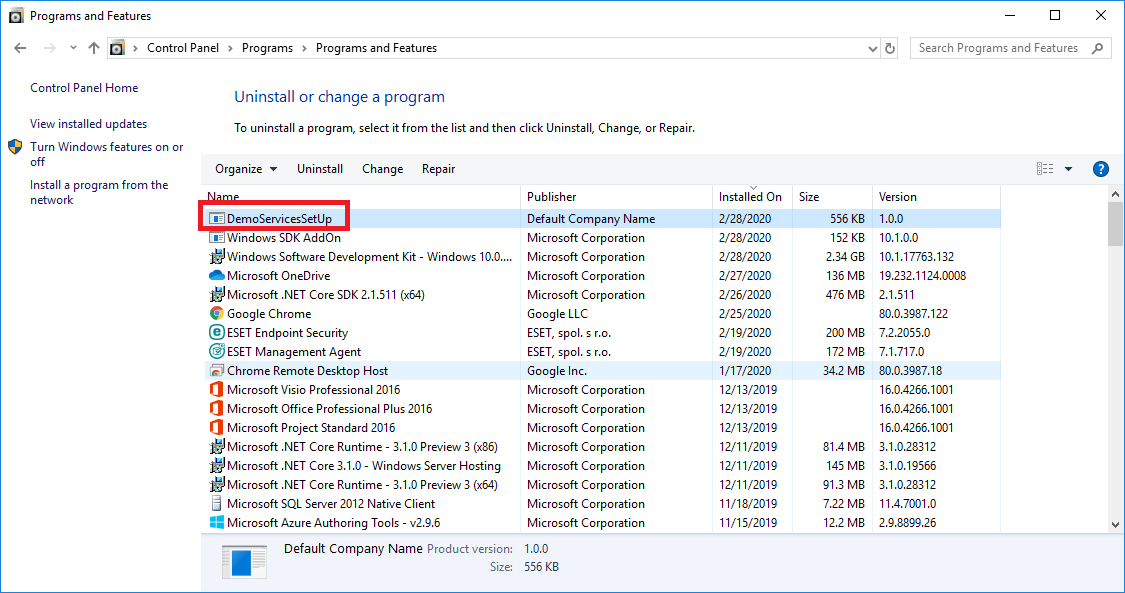


# Gỡ cài đặt (Uninstalling)

Để gỡ Services, bạn vào **Control Panel🡪 Uninstall Progames**



Sau khi cửa sổ **Uninstall** hiện lên, bạn có thể thấy **Progam** tên là **DemoServicesSetup**



Click chuột phải vào **DemoServicesSetup** và chọn **Uninstall** để gỡ **Services.**

Sau khi gỡ cài đặt thành công bạn có thể mở cửa sổ **Services** lên và bấm **Refresh** kiểm tra lại.

