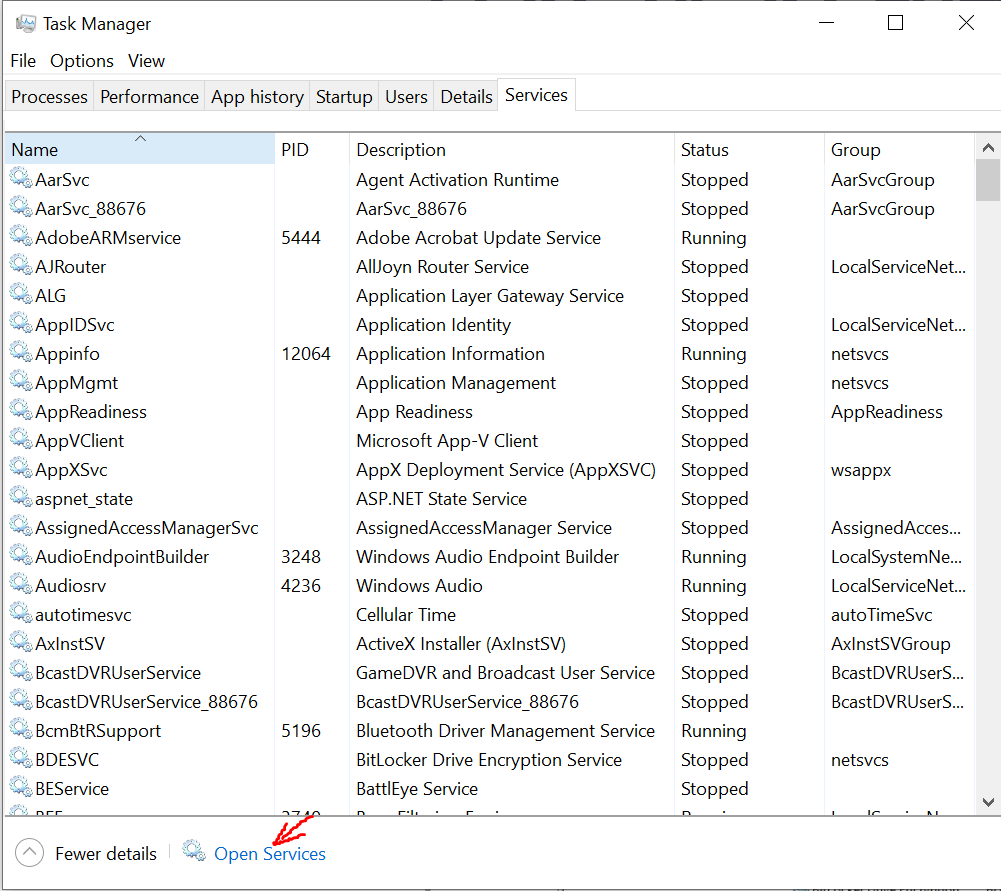
**Usługi Windows – tworzenie i instalacja**

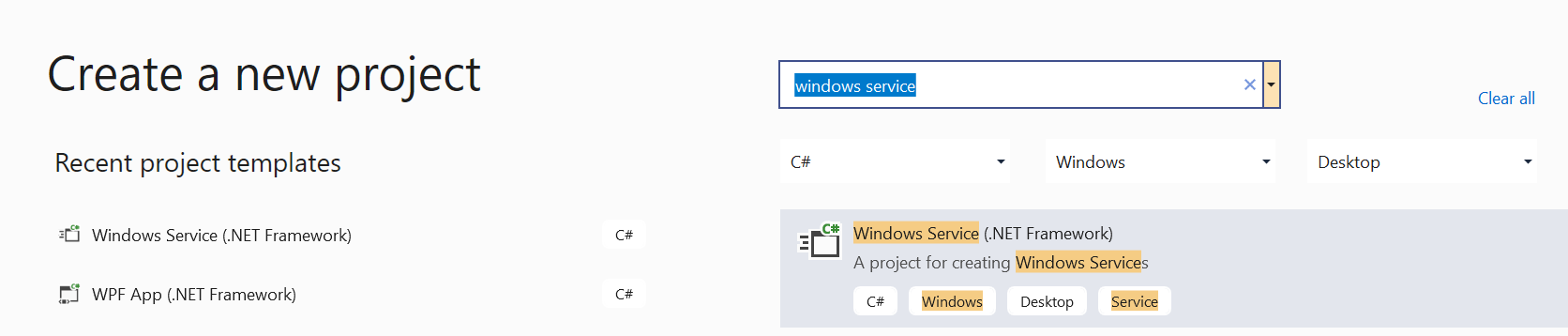
Usługa Windows, to proces, który jest wykonywany w obrębie systemu operacyjnego Windows. Nie posiada on własnego interfejsu użytkownika,  
nie podejmuje żadnej interakcji z użytkownikiem, działa w tle i wykonuje różne zadania.  
Działa od razu po uruchomieniu systemu operacyjnego.

Ponieważ usługa nie ma żadnego interfejsu, to jedynym, w którym możemy ją  
znaleźć jest okno Usług Windows.

Aby przejść do usług wpisujemy: **services.msc**,  
lub z poziomu menedżera zadań w zakładce Services



**Tworzenie Usługi Windows w Visual Studio**



Zostaje utworzony szkielet aplikacji  
Klasę **Program.cs** a w niej metodę **Main()**, która startuje naszą usługę.

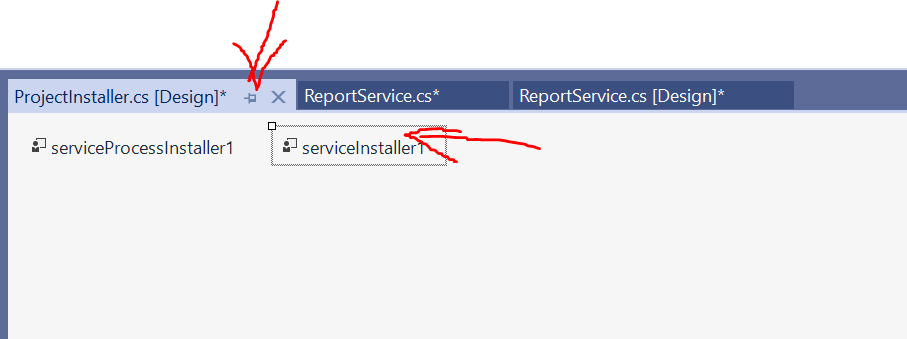
Następnie klasa **Service.cs**. Jeżeli chcemy zobaczyć jej kod to  
wciskamy **F7** lub z menu kontekstowego wybieramy: **View Code**.

Metoda ma konstruktor, w nim inicjalizację komponentów  
oraz dwie metody: **OnStart()** wywoływana przy starcie usługi i **OnStop()** wywoływana przy zatrzymaniu usługi.

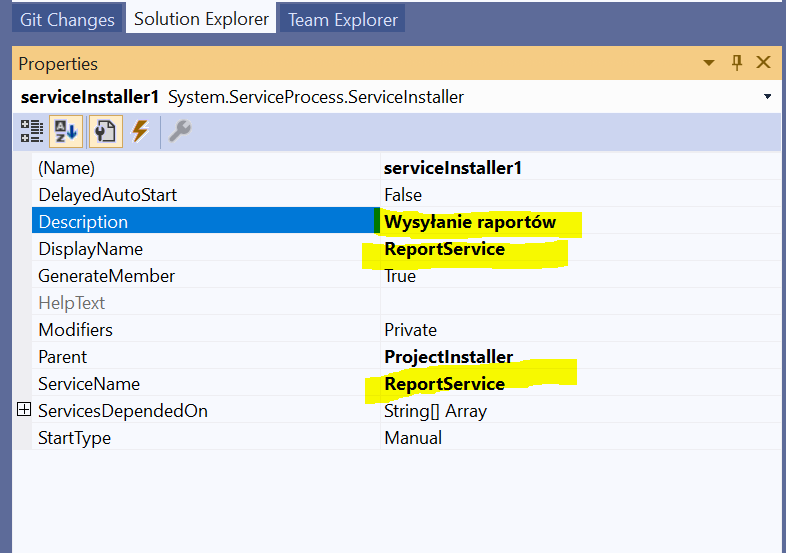
Zmieniamy nazwę tej klasy z Service.cs na nazwę adekwatną do jej funkcjonalności np. **ReportService.cs**

**Dodajemy Installera**

Po dwukrotnym kliknięci na Service.cs pojawia się okno designera.  
Na oknie Designer’a wybieramy z menu **Add Installer**, zostaje dodany plik   
**ProjectInstaller.cs**.

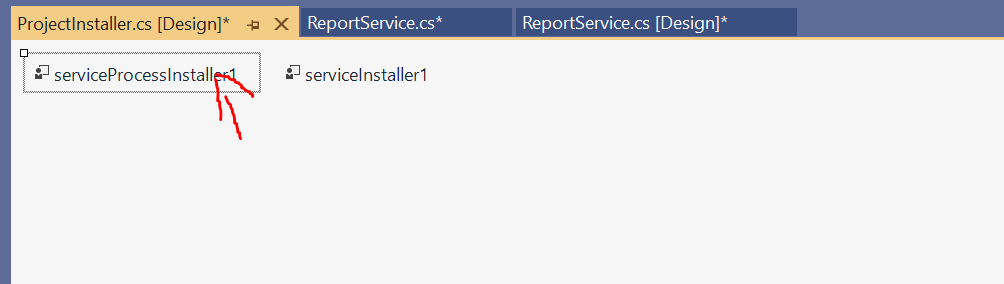


Klikamy na serviceInstaller1 i zmieniamy nazwy ServiceName, DisplayName i Description.

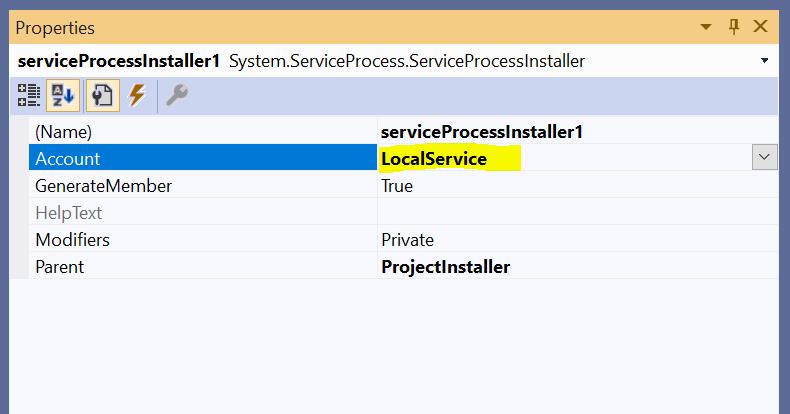


Dane te będą wyświetlane w oknie gdzie możesz zarządzać usługami.

Klikamy jeszcze na **serviceProcessInstaller1**



i zmieniamy przy opcji **Account** na **LocalService** aby usługa uruchamiała się dla wszystkich użytkowników.



**Dodajemy trochę kodu ułatwiającego uruchomienie usługi**

Modyfikujemy metodę Main w klasie program

|  |
| --- |
| static void Main(string[] args)  {  ServiceBase[] ServicesToRun;  ServicesToRun = new ServiceBase[]  {  new ReportService()  };  ServiceBase.Run(ServicesToRun);  } |

Kod:

|  |
| --- |
| ServiceBase[] ServicesToRun;  ServicesToRun = new ServiceBase[]  {  new ReportService()  };  ServiceBase.Run(ServicesToRun); |

Zastępujemy kodem:

|  |
| --- |
| if (Environment.UserInteractive)  {  var parameter = string.Concat(args);  switch (parameter)  {  case "--install":  ManagedInstallerClass.InstallHelper(new[]  {Assembly.GetExecutingAssembly().Location});  break;  case "--uninstall":  ManagedInstallerClass.InstallHelper(new[]  {"/u",Assembly.GetExecutingAssembly().Location});  break;  default:  break;  }  }  else  {  ServiceBase[] ServicesToRun;  ServicesToRun = new ServiceBase[]  {  new ReportService()  };  ServiceBase.Run(ServicesToRun);  } |

Dodajemy niezbędne using’i.

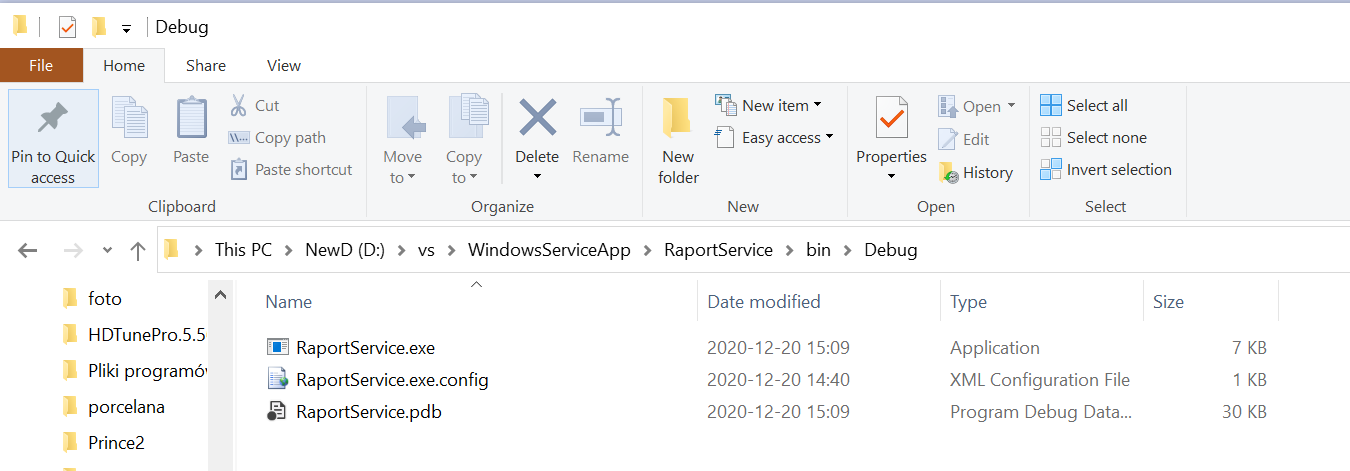
Po powyższych zmianach kompilujemy program.

Po tych modyfikacjach nasza usługa nadaje się już do instalacji

**Instalowanie usługi w Windows**

Na projekcie klikamy **open in file explorer**, przechodzimy do bin, następnie do debug w celu skopiowania ścieżki dostępu do pliku.

Poniżej zainstalowanie przykładowej usługi



Wklejamy ścieżkę, dopisujemy nazwę naszego exeka i dopisujemy –install

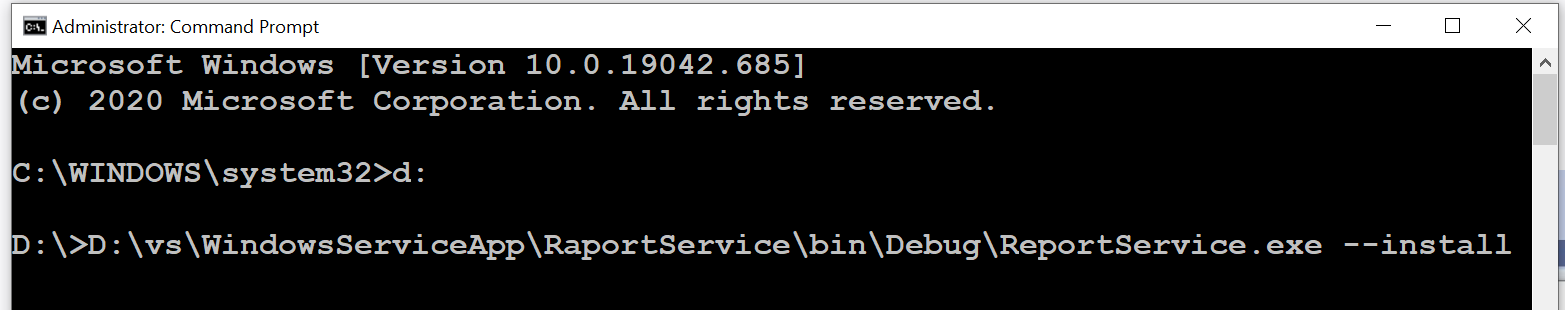
**Instalacja i deinstalacja usługi ReportServices**

|  |
| --- |
| D:\vs\WindowsServiceApp\ReportService\bin\Debug\ReportService.exe --install D:\vs\WindowsServiceApp\ReportService\bin\Debug\ReportService.exe --uninstall |

**Instalacja i deinstalacja usługi SmsNotification**

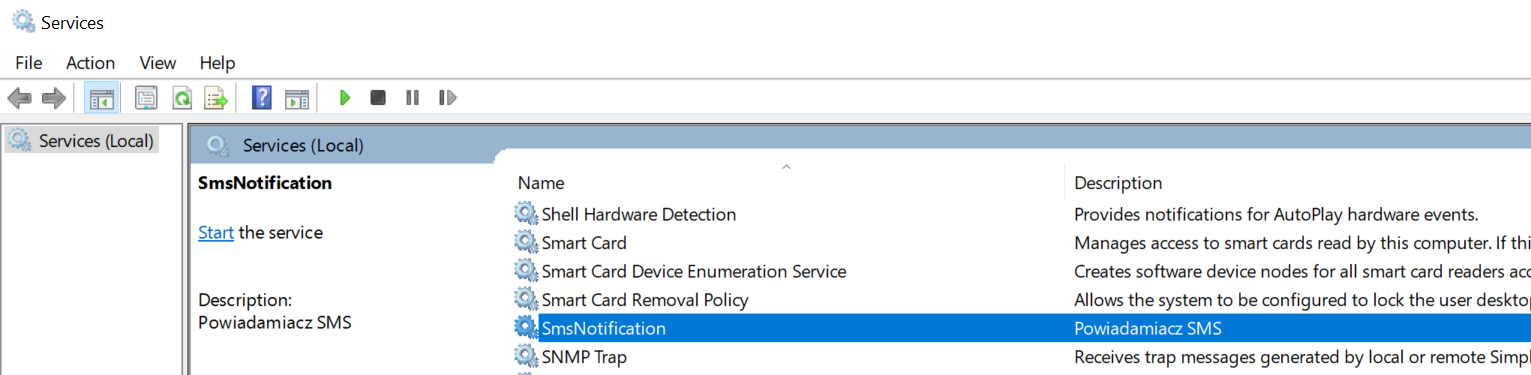
|  |
| --- |
| D:\vs\WindowsServiceApp\SmsNotification\bin\Debug\SmsNotification.exe --install D:\vs\WindowsServiceApp\SmsNotification\bin\Debug\SmsNotification.exe --uninstall |

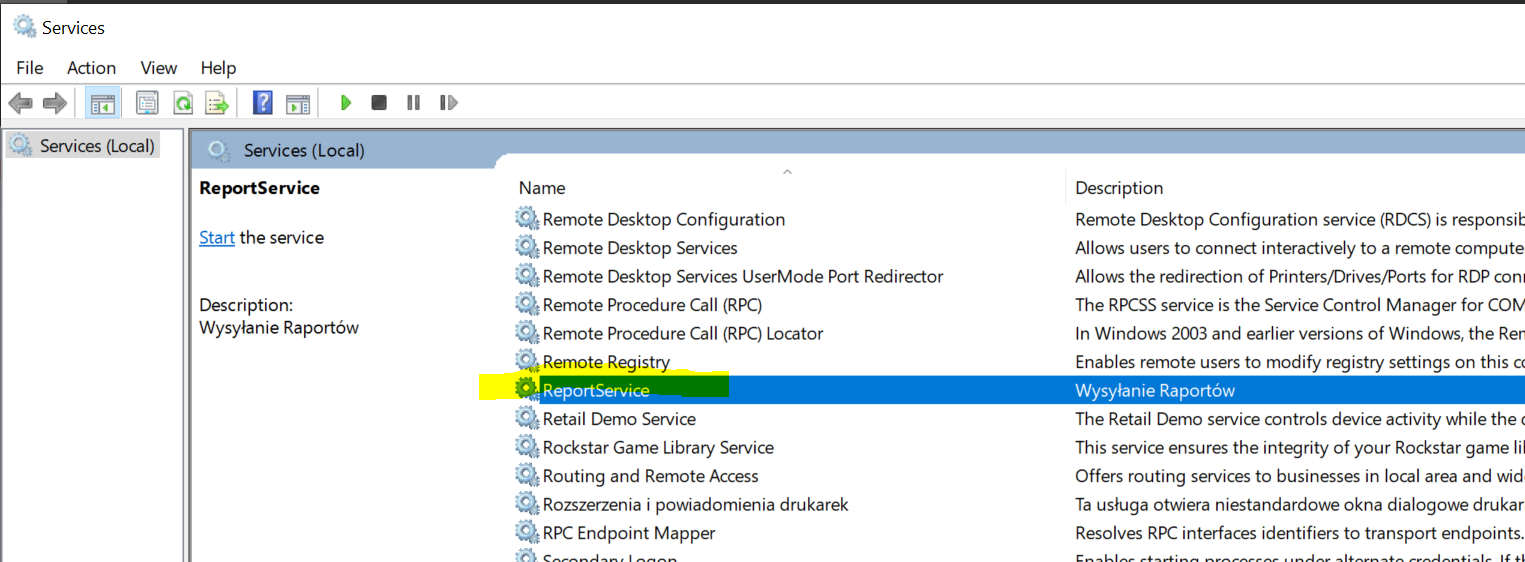
Uruchamiamy cmd **z uprawnieniami administratora**



Po instalacji powstaje plik : **ReportService.InstallLog**

Po instalacji zainstalowane usługi widoczne w services.msc





**Wdrażanie Usługi Windows**

Przebudowujemy aplikację w trybie **release**  
Kopiujemy pliki:

