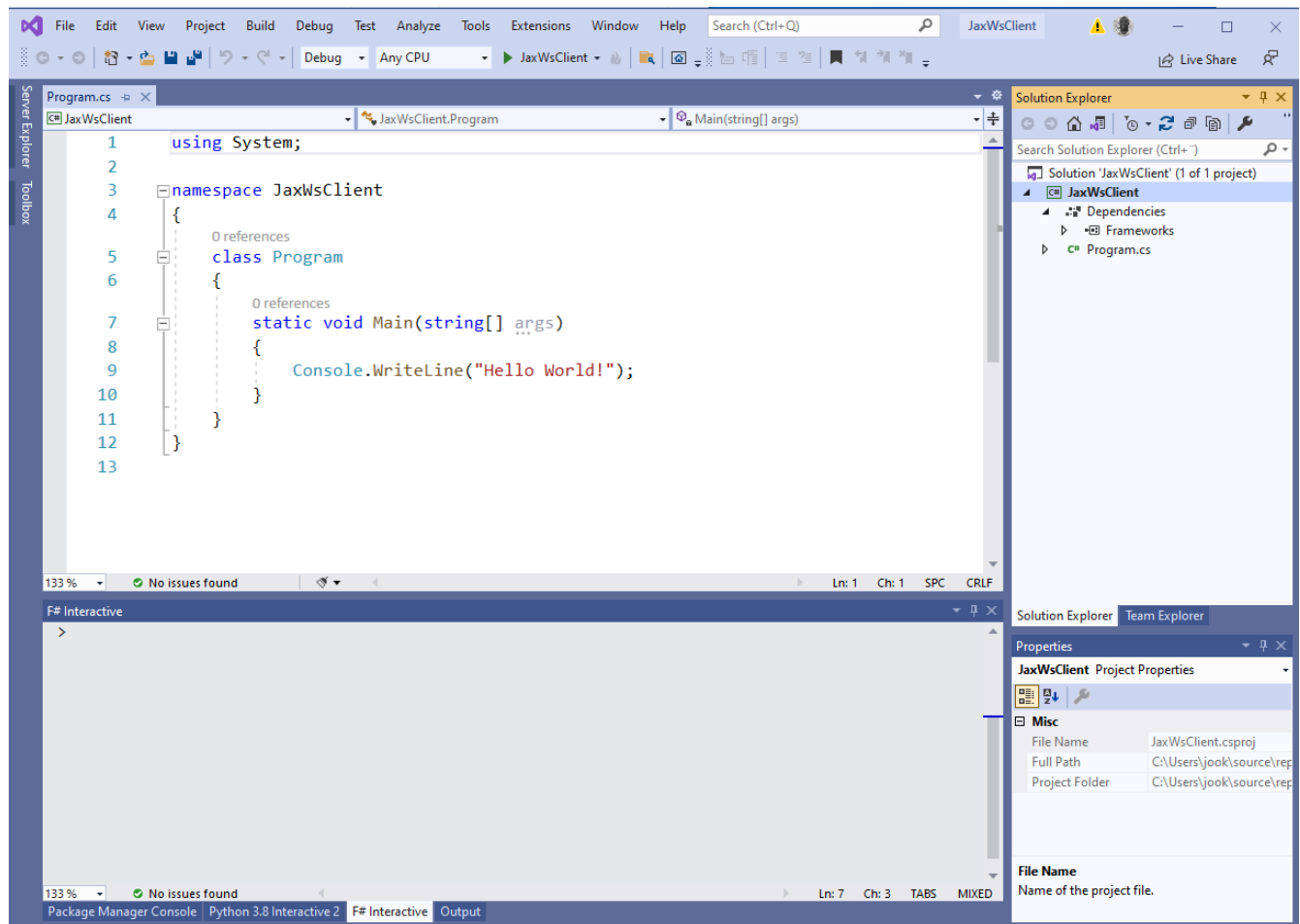


C# .NET Client to Java JAX-WS SOAP Web Service

In this tutorial, I will show you how to use Visual Studio (VS22) and Windows Communication Foundation (<https://docs.microsoft.com/en-us/dotnet/framework/wcf/whats-wcf>) to consume a Java SOAP web service.

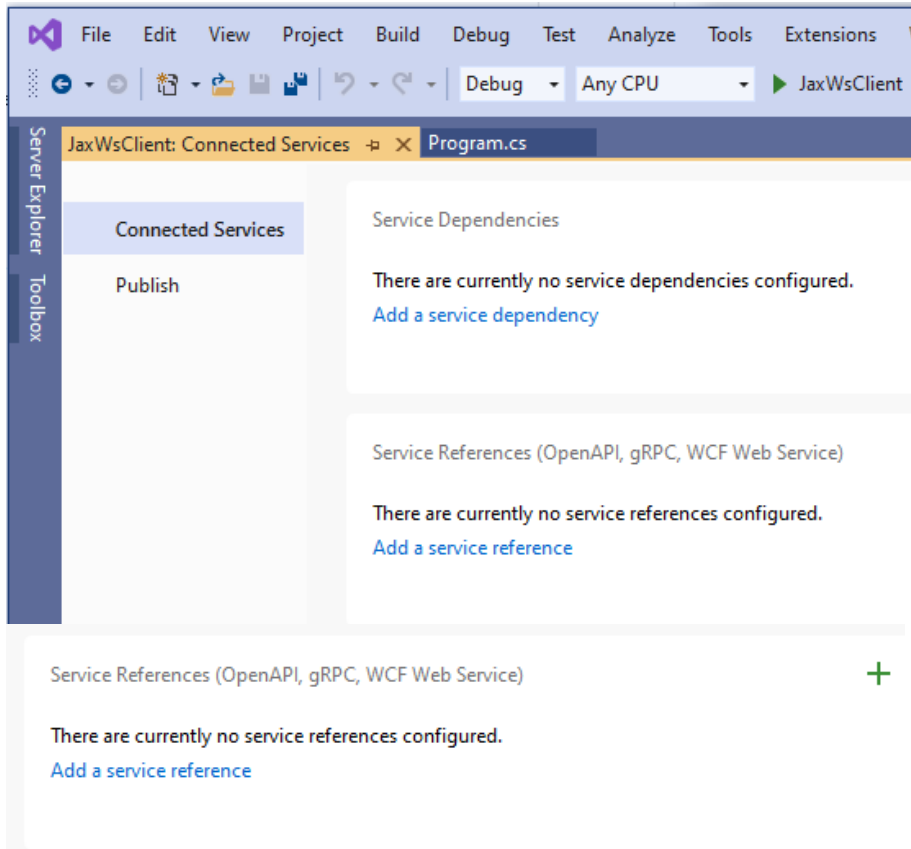
1) Create a console application



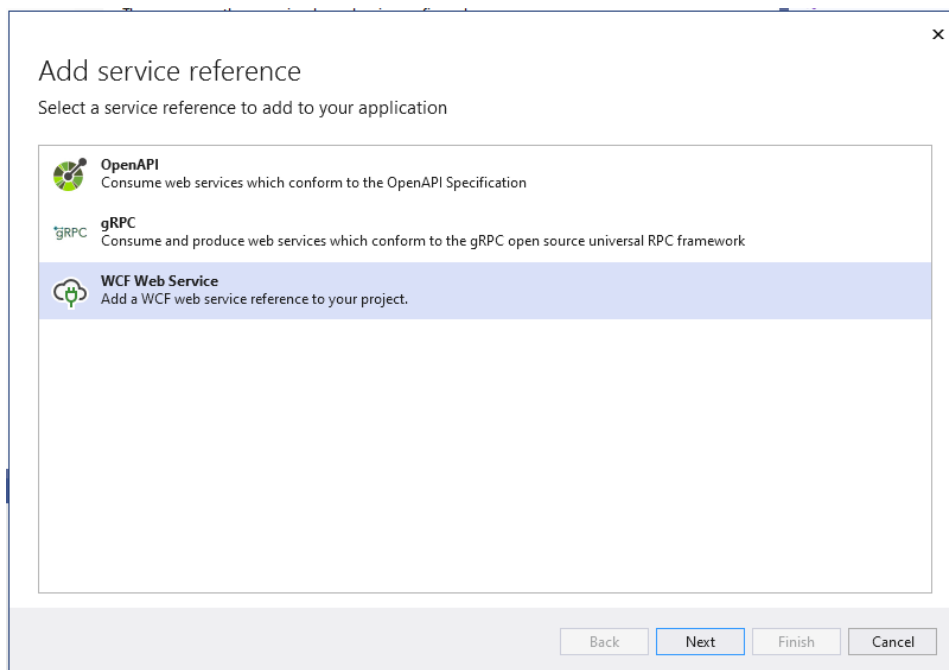
In Solution Explorer, double-click the **Connected Services** node of the project (for a .NET Core or .NET Standard project this option is available when you right-click on the **Dependencies** node of the project in Solution Explorer).

2) Connected Services (WCF Web Service)

The **Connected Services** page appears as shown in the following image:



Click Add a service reference



Click Next

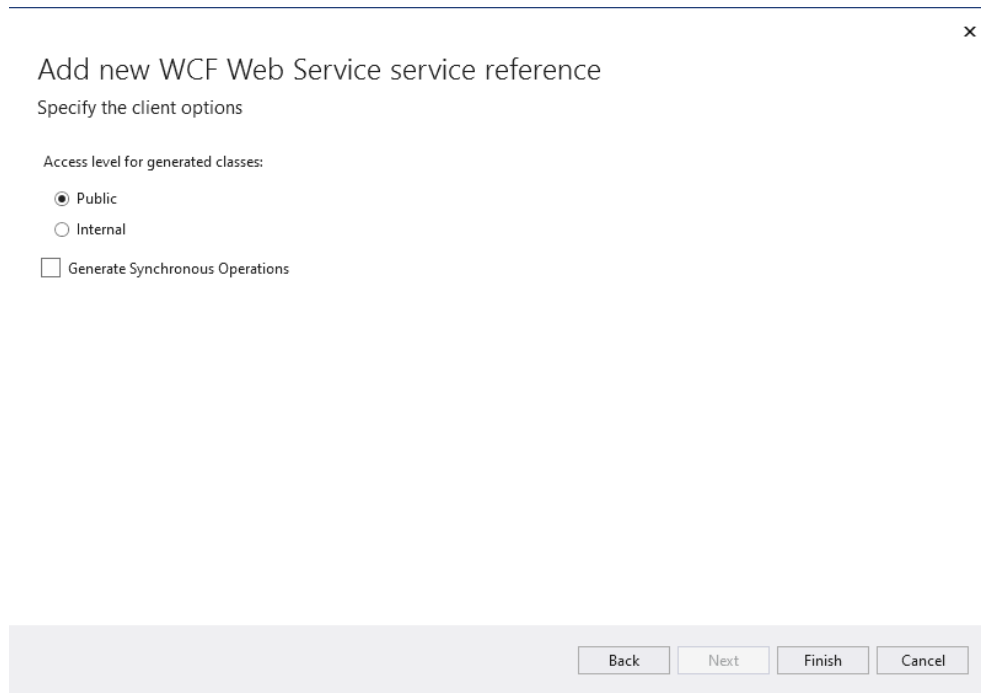
3) Service Reference

The dialog box is titled "Add new WCF Web Service service reference". Below the title, it says "Specify the service to add". A paragraph of instructions follows: "To see a list of available services on a specific server, enter a service URL and select Go. To find available services in the solution, select Discover. To load a service metadata from a WSDL file, select Browse." Below this is a "URI:" label and a text box containing "http://localhost:9090/jaxws-wakeup-service?wsdl". To the right of the text box are three buttons: "Stop", "Discover", and "Browse...". Below these are two panels: "Services:" and "Operations:". The "Services:" panel shows a tree view with "WakeUpMessageService" expanded, revealing a sub-item "WakeUpMessageService". The "Operations:" panel is empty. Below the panels is a "Status:" label and a text box containing "Number of services found: 1". Below that is a "Namespace:" label and a text box containing "ServiceReference1". At the bottom are four buttons: "Back", "Next" (highlighted with a blue border), "Finish", and "Cancel".

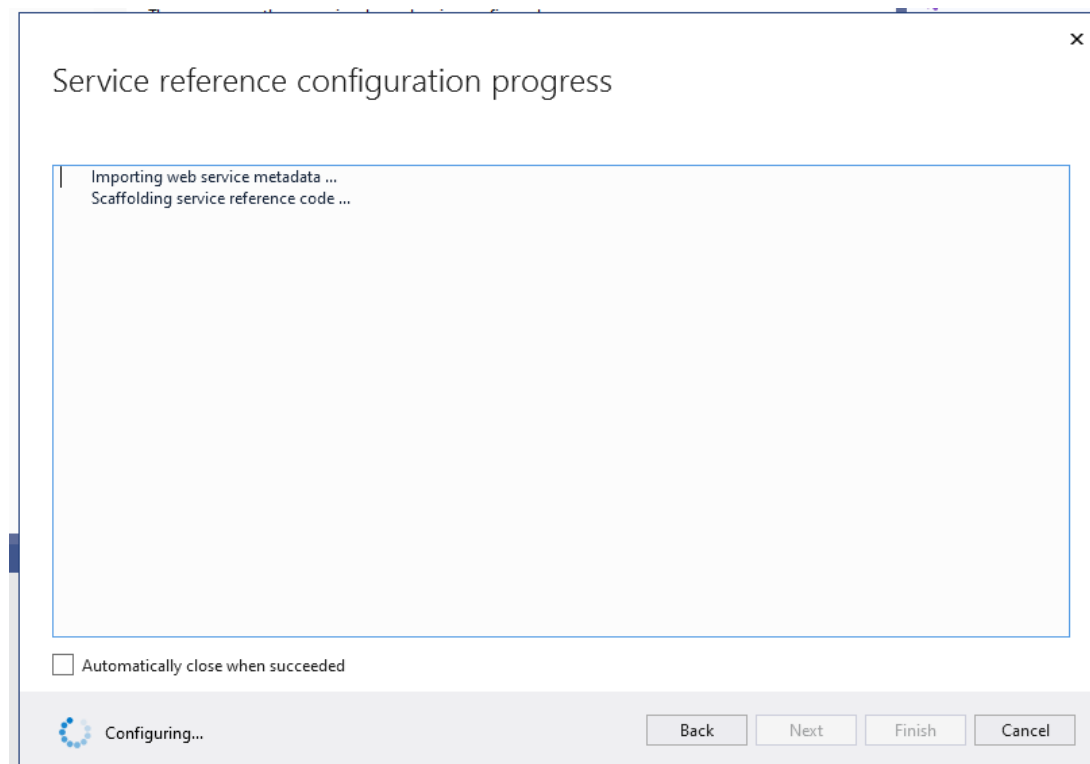
Click Next

The dialog box is titled "Add new WCF Web Service service reference". Below the title, it says "Specify the data type options". There are three options: "Always generate message contracts" (unchecked), "Collection type:" (a dropdown menu showing "System.Array"), and "Dictionary collection type:" (a dropdown menu showing "System.Collections.Generic.Dictionary"). Below these is a checked checkbox "Reuse types in referenced assemblies". Under this checkbox are two radio buttons: "Reuse types in all referenced assemblies" (selected) and "Reuse types in specified referenced assemblies:". Below the radio buttons is a large empty text box. At the bottom are four buttons: "Back", "Next" (highlighted with a blue border), "Finish", and "Cancel".

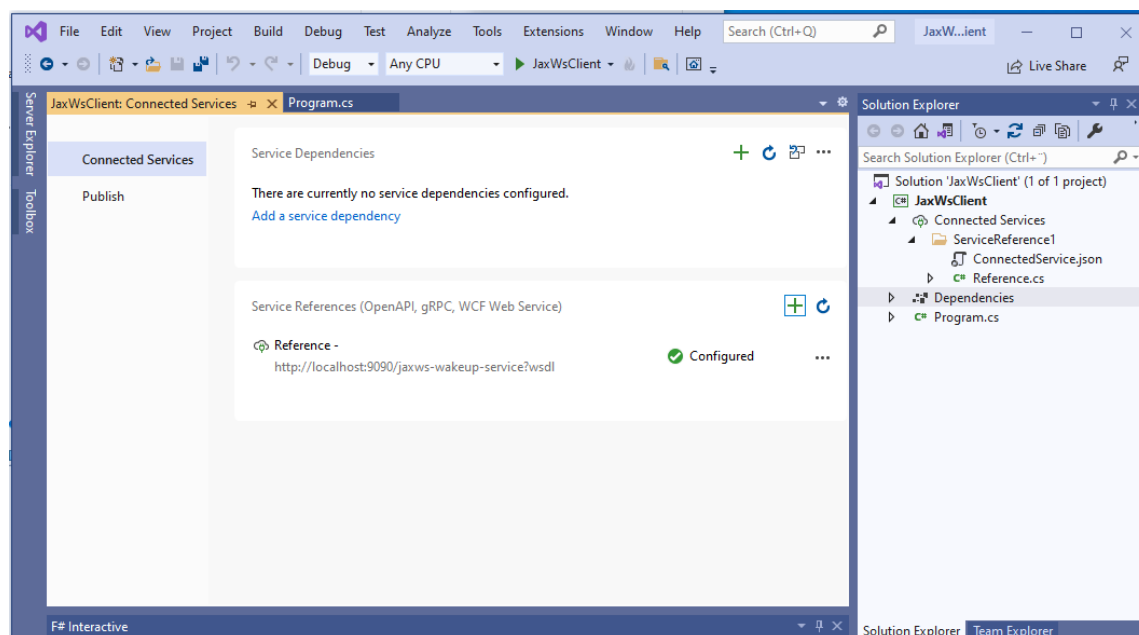
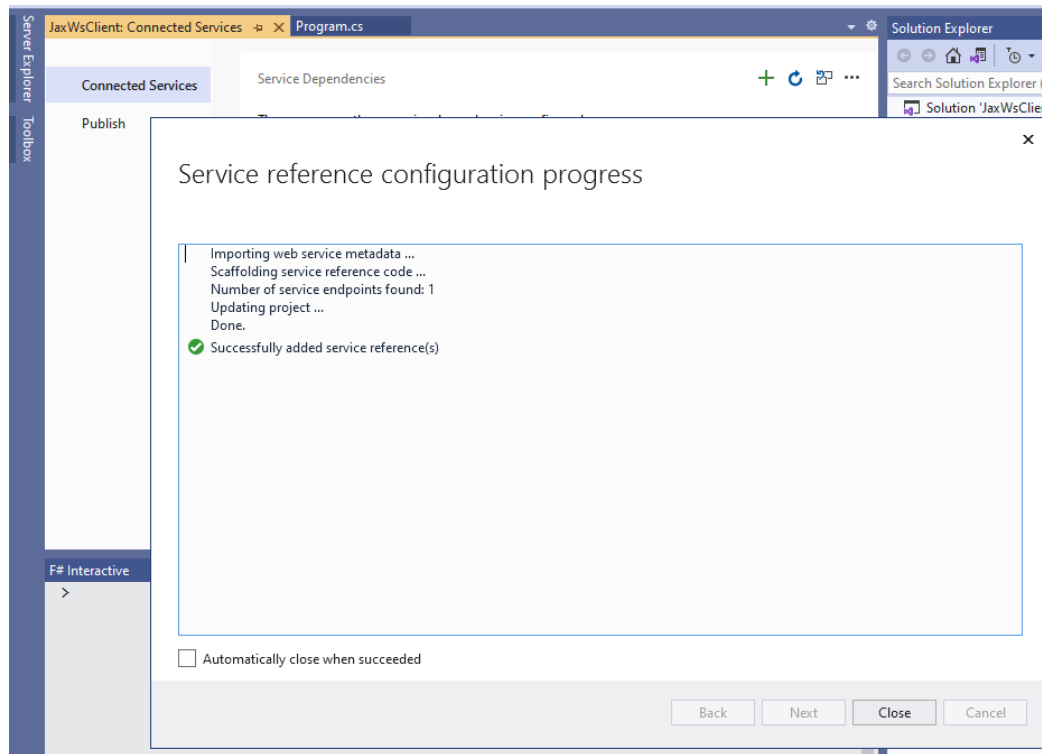
Click Next, and Select Public in the next window and the click Finish



Visual Studio configures the service reference that can be used in the code.



When done click Close as shown in the next screenshot



You can now see the generated service reference codes in the Solution Explorer.

The next step is to write the client code that utilizes this generated service reference.
The code snippet is shown in the next page.

4) Write your code

After the service reference configuration, import the service reference (ServiceReference1) from step 3, adding the namespace in the console app as shown below. Now you can call the web service methods from the console app.

```
using System;
using ServiceReference1;

namespace JaxWsClient
{
    class Program
    {
        static void Main()
        {
            Console.WriteLine("*** Calling JAX-WS WakeUpMessageService");
            ServiceReference1.WakeUpMessageServiceClient
                client = new WakeUpMessageServiceClient();
            System.Threading.Tasks.Task<wakeUpResponse>
                result1 = client.wakeUpAsync("Alfo Ben Khaledy");
            string r2 = result1.Result.Body.@return;
            Console.WriteLine("**** Result **** \n" + r2);
            client.Close();
        }
    }
}
```

5) Test your code

Set the console application project as the StartUp project by a right-click.

Right click JaxWsClient in the solution explorer > Set as Startup Project

Run the Program

