Soap Web Service Using APL64 Cross-Platform Component

Contents

Overview

- Create a SOAP web service with algorithmic support provided by an APL64 cross-platform component.
- Access the SOAP web service via a SOAP browser client.
- A SOAP web service uses the Simple Object Access Protocol to specify messages which can be transmitted between a web server and web server client. The SOAP protocol is inherently cross-platform because message content is rendered in XML format. XML format content is self-documenting since it is composed of descriptive tags and data enclosed within tag delimiters.

SOAP Message Structure

A SOAP message is an XML-format document containing these elements:

- Envelope identifying the XML document as a SOAP message
- Header containing 'header' information
- Body containing request or response information
- Fault containing error and status information

SOAP Web Service Example

This document provides instructions for creating a simple SOAP web service using the http protocol to transmit SOAP messages between a web client and web server.

- The web service is created in Microsoft Visual Studio using the C# programming language
- The web service can be hosted on .Net compatible platforms such as Windows and Linux
- The functionality of the web service is provided by an APL64 cross-platform component

Download the Example Source Code

Download the file: https://aplnow.com/APL64/UserDocumentation/APL64_CPC_IN_SOAP_SVC.zip

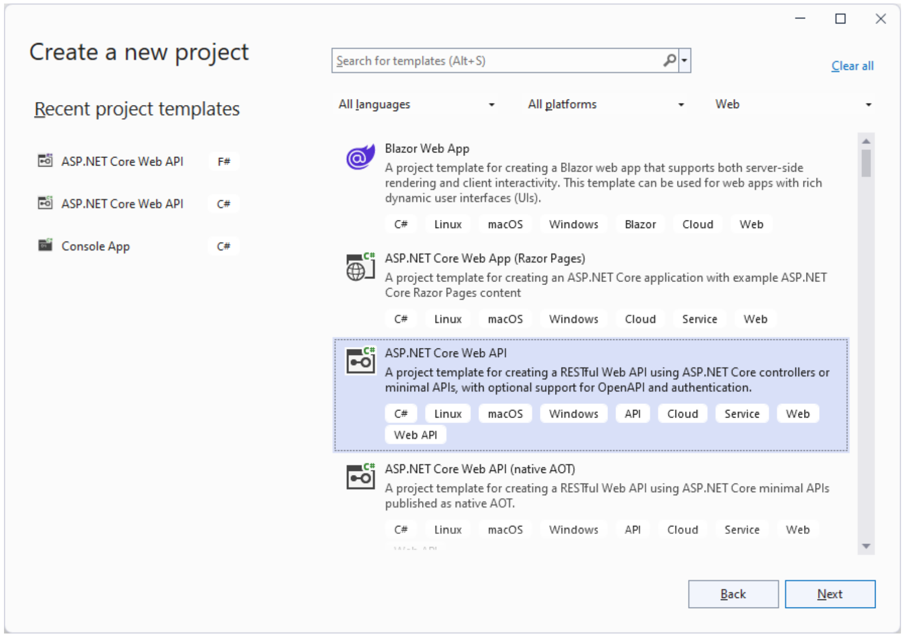
Extract the file to 'c:\'

Follow the instructions in the remainder of this document to create the APL64 CPC and run the solution in Visual Studio.

SOAP Web Service Visual Studio Project

Create the SoapWebService Project in Visual Studio

Select the ASP.Net Core Web API project template for C#:



Create a new project

Search for templates (Alt+S)

Clear all

Recent project templates

ASP.NET Core Web API	F#
ASP.NET Core Web API	C#
Console App	C#

All languages | All platforms | Web

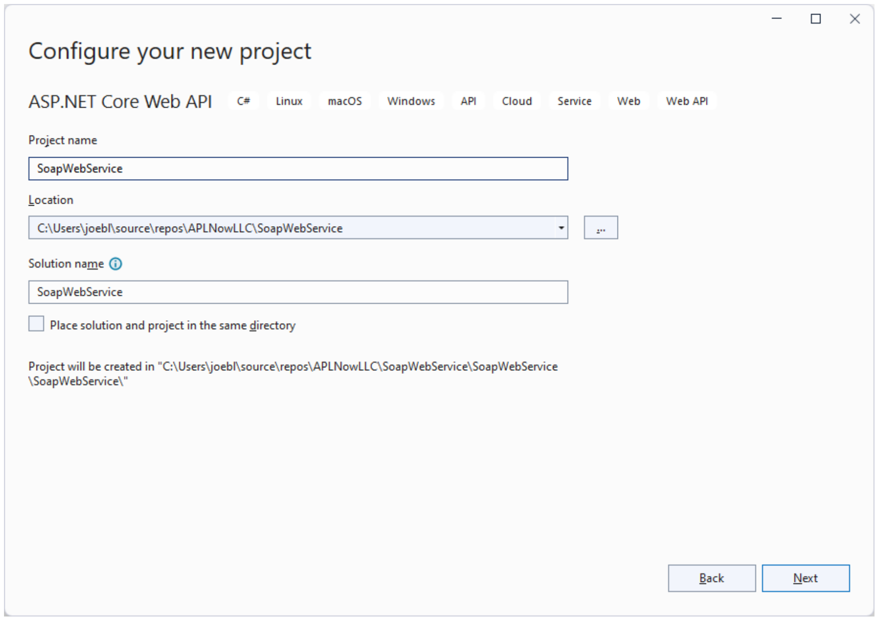
Blazor Web App
A project template for creating a Blazor web app that supports both server-side rendering and client interactivity. This template can be used for web apps with rich dynamic user interfaces (UIs).
C# Linux macOS Windows Blazor Cloud Web

ASP.NET Core Web App (Razor Pages)
A project template for creating an ASP.NET Core application with example ASP.NET Core Razor Pages content
C# Linux macOS Windows Cloud Service Web

ASP.NET Core Web API
A project template for creating a RESTful Web API using ASP.NET Core controllers or minimal APIs, with optional support for OpenAPI and authentication.
C# Linux macOS Windows API Cloud Service Web Web API

ASP.NET Core Web API (native AOT)
A project template for creating a RESTful Web API using ASP.NET Core minimal APIs published as native AOT.
C# Linux macOS Windows API Cloud Service Web

Configure the SoapWebService project



Project name: SoapWebService

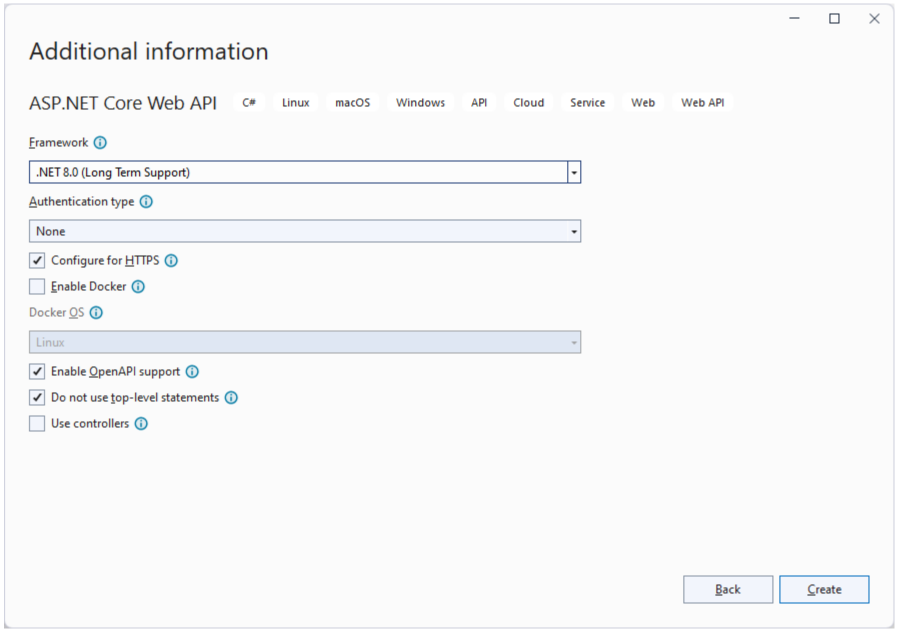
Location: C:\Users\joebl\source\repos\APLNowLLC\SoapWebService

Solution name: SoapWebService

☐ Place solution and project in the same directory

Project will be created in "C:\Users\joebl\source\repos\APLNowLLC\SoapWebService\SoapWebService\SoapWebService\"

Supply the Additional Project Information



Framework: .NET 8.0 (Long Term Support)

Authentication type: None

☑ Configure for HTTPS

☐ Enable Docker

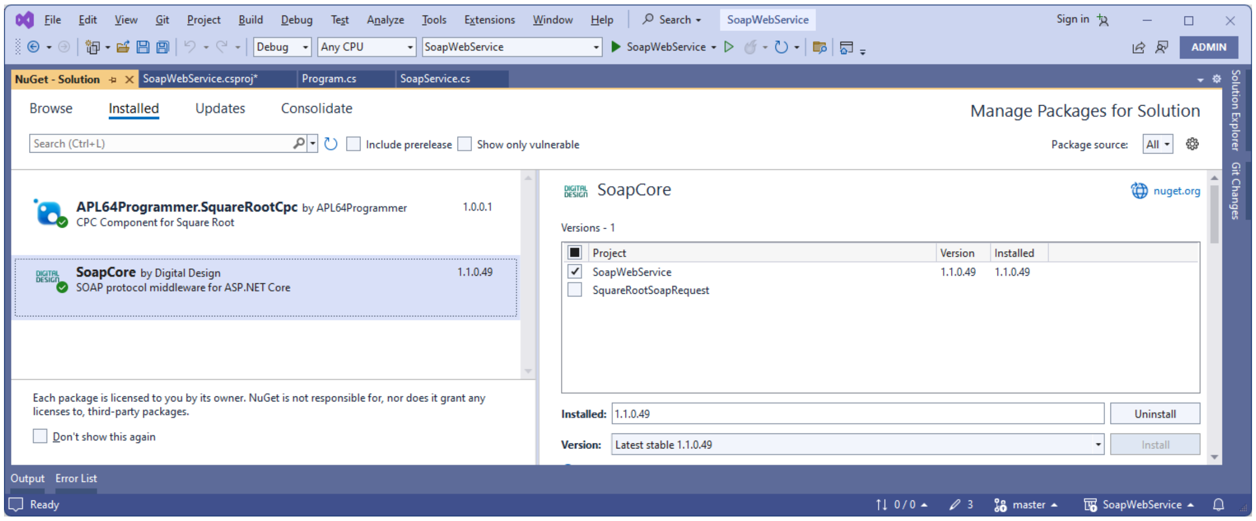
Docker OS: Linux

☑ Enable OpenAPI support

☑ Do not use top-level statements

☐ Use controllers

Install SoapCore NuGet package into the SoapWebService project



Manage Packages for Solution

	Version	Installed
APL64Programmer.SquareRootCpc by APL64Programmer — CPC Component for Square Root	1.0.0.1	
SoapCore by Digital Design — SOAP protocol middleware for ASP.NET Core	1.1.0.49	

SoapCore — Versions - 1

Project	Version	Installed
✓ SoapWebService	1.1.0.49	1.1.0.49
SquareRootSoapRequest		

Installed: 1.1.0.49

Version: Latest stable 1.1.0.49

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

Remove Unnecessary Project Template Information

Close the ‘ASP.Net Core’ introductory page tab.

Delete the WeatherForecast.cs code file from the project

Delete the Properties folder from the project

Uninstall the Swashbuckle.AspNetCore Nuget package

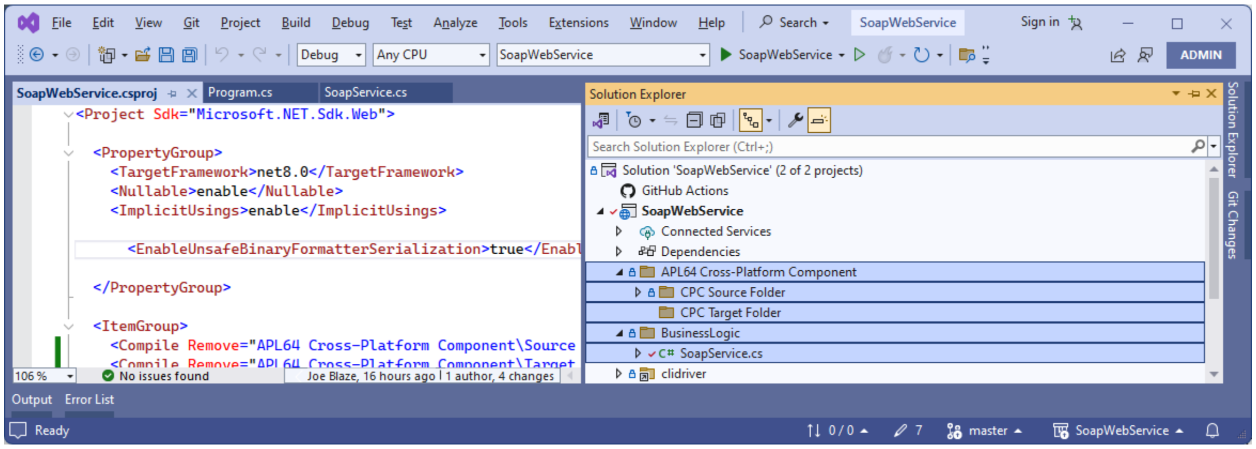
Uninstall the Microsoft.AspNetCore.OpenApi Nuget package.

Create Project Folders

‘BusinessLogic’ folder

‘APL64 Cross-Platform Component’ folder and sub-folders:

- ‘CPC Source Folder’
- ‘CPC Target Folder’

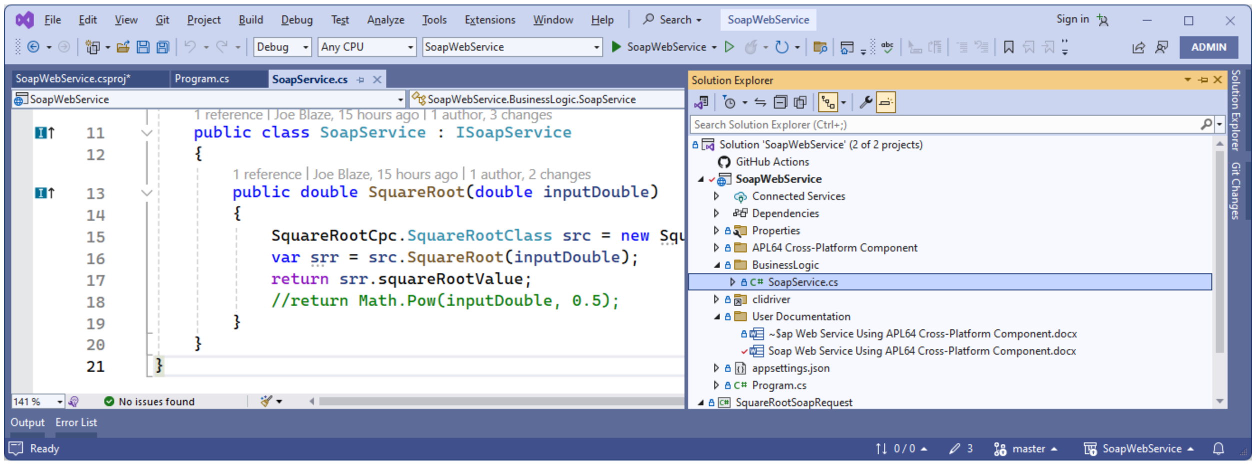


```
<Project Sdk="Microsoft.NET.Sdk.Web">

  <PropertyGroup>
    <TargetFramework>net8.0</TargetFramework>
    <Nullable>enable</Nullable>
    <ImplicitUsings>enable</ImplicitUsings>

      <EnableUnsafeBinaryFormatterSerialization>true</Enabl

  </PropertyGroup>

  <ItemGroup>
    <Compile Remove="APL64 Cross-Platform Component\Source
    <Compile Remove="APL64 Cross-Platform Component\Target
```

```
Solution 'SoapWebService' (2 of 2 projects)
  GitHub Actions
  SoapWebService
    Connected Services
    Dependencies
    APL64 Cross-Platform Component
      CPC Source Folder
      CPC Target Folder
    BusinessLogic
      SoapService.cs
  clidriver
```

Create SoapService Class

Create a SoapService.cs code file in the BusinessLogic folder:

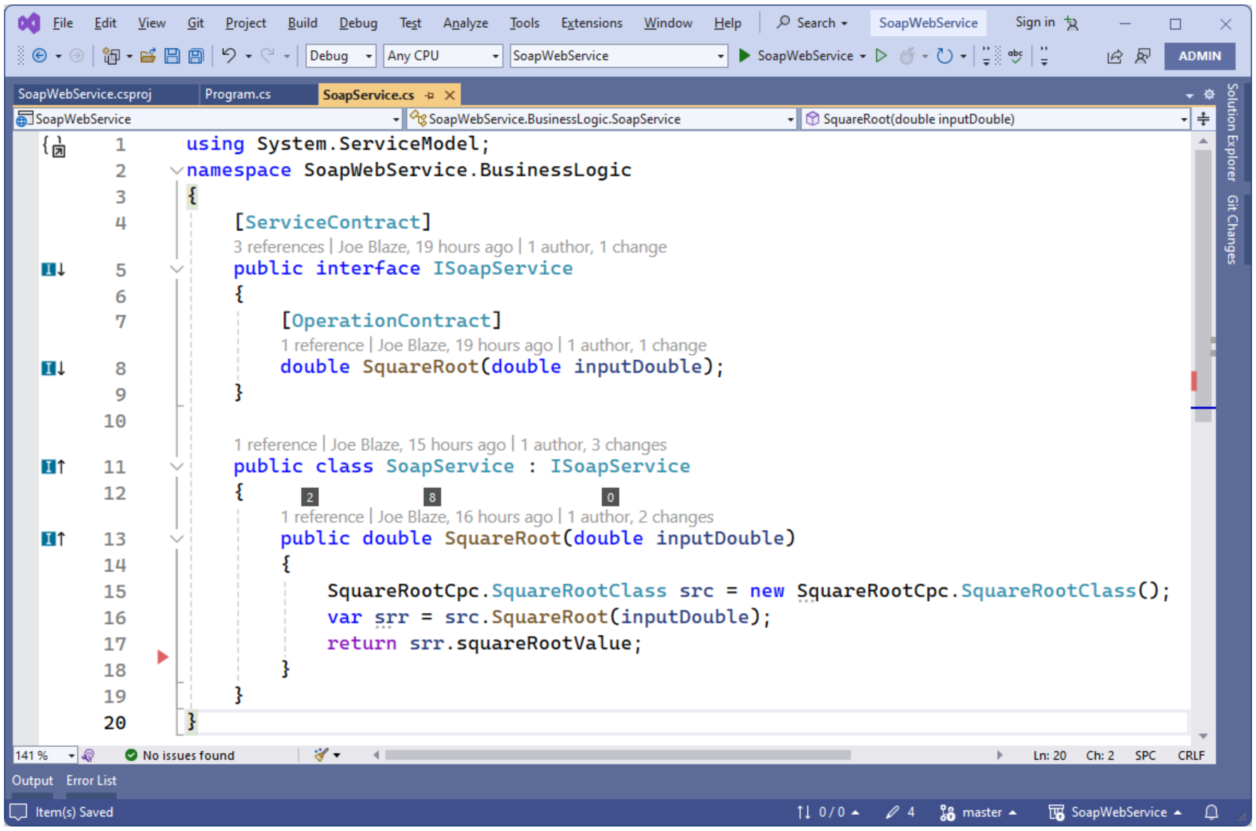


```
1 reference | Joe Blaze, 15 hours ago | 1 author, 3 changes
public class SoapService : ISoapService
{
    1 reference | Joe Blaze, 15 hours ago | 1 author, 2 changes
    public double SquareRoot(double inputDouble)
    {
        SquareRootCpc.SquareRootClass src = new Squ
        var srr = src.SquareRoot(inputDouble);
        return srr.squareRootValue;
        //return Math.Pow(inputDouble, 0.5);
    }
}
```

```
Solution 'SoapWebService' (2 of 2 projects)
    GitHub Actions
  SoapWebService
      Connected Services
      Dependencies
      Properties
      APL64 Cross-Platform Component
      BusinessLogic
        SoapService.cs
      clidriver
      User Documentation
        ~$ap Web Service Using APL64 Cross-Platform Component.docx
        Soap Web Service Using APL64 Cross-Platform Component.docx
      appsettings.json
      Program.cs
  SquareRootSoapRequest
```

Replace the code in the SoapService.cs code file with this code:

```
using System.ServiceModel;
namespace SoapWebService.BusinessLogic
{
    [ServiceContract]
    public interface ISoapService
    {
        [OperationContract]
        double SquareRoot(double inputDouble);
    }

    public class SoapService : ISoapService
    {
        public double SquareRoot(double inputDouble)
        {
            SquareRootCpc.SquareRootClass src = new SquareRootCpc.SquareRootClass();
            var srr = src.SquareRoot(inputDouble);
            return srr.squareRootValue;
        }
    }
}
```

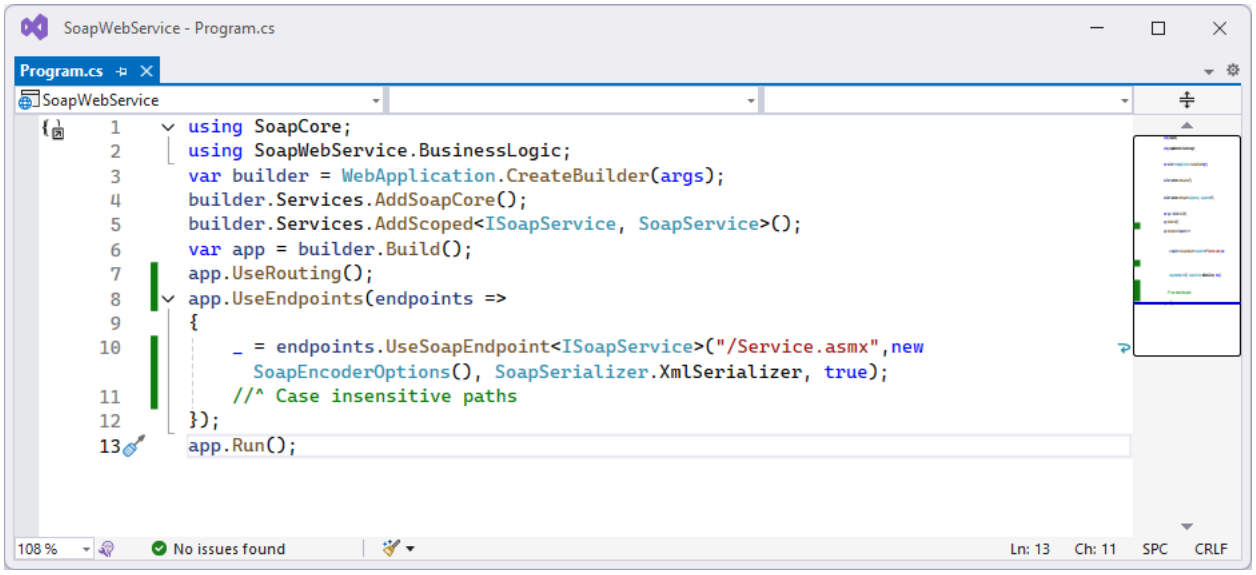


SoapWebService.csproj
Program.cs
SoapService.cs
SoapWebService
SoapWebService.BusinessLogic.SoapService
SquareRoot(double inputDouble)

```
using System.ServiceModel;
namespace SoapWebService.BusinessLogic
{
    [ServiceContract]
    3 references | Joe Blaze, 19 hours ago | 1 author, 1 change
    public interface ISoapService
    {
        [OperationContract]
        1 reference | Joe Blaze, 19 hours ago | 1 author, 1 change
        double SquareRoot(double inputDouble);
    }

    1 reference | Joe Blaze, 15 hours ago | 1 author, 3 changes
    public class SoapService : ISoapService
    {
        1 reference | Joe Blaze, 16 hours ago | 1 author, 2 changes
        public double SquareRoot(double inputDouble)
        {
            SquareRootCpc.SquareRootClass src = new SquareRootCpc.SquareRootClass();
            var srr = src.SquareRoot(inputDouble);
            return srr.squareRootValue;
        }
    }
}
```

Replace the code in the Program.cs code file

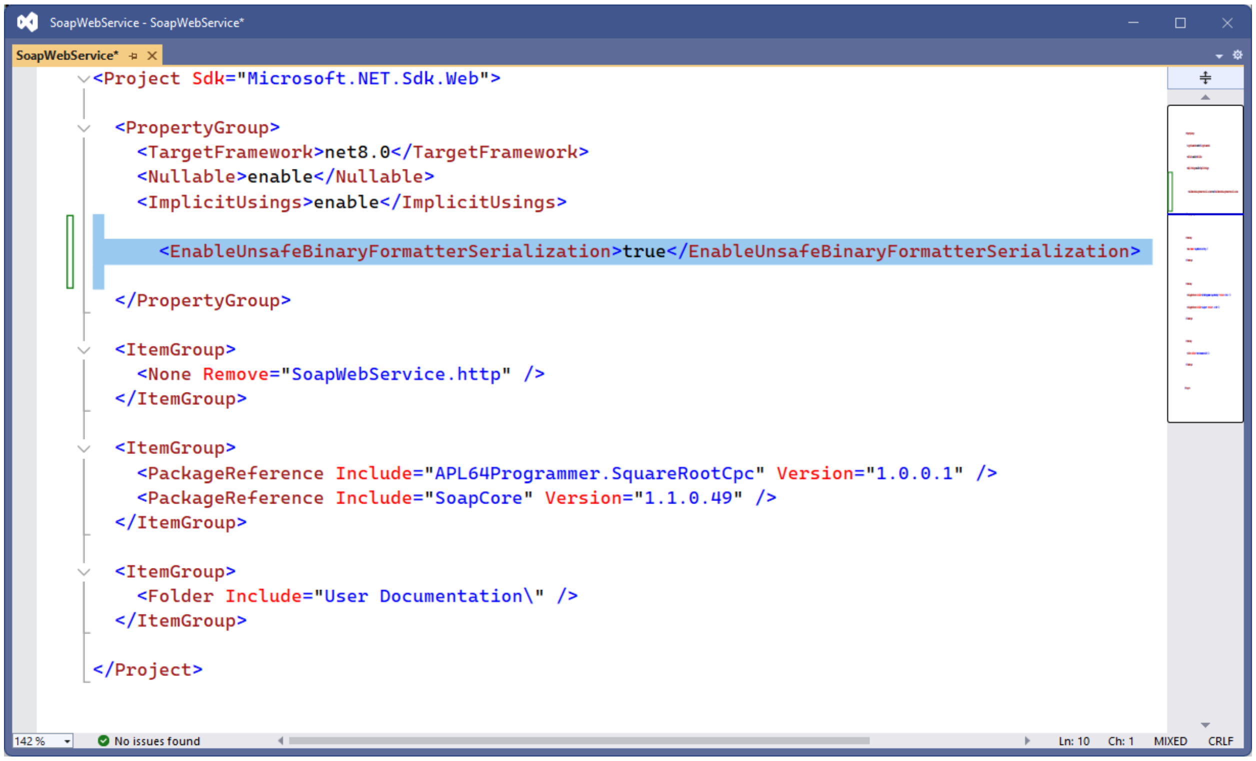
```
using SoapCore;
using SoapInNetCore.BusinessLogic;
var builder = WebApplication.CreateBuilder(args);
builder.Services.AddSoapCore();
builder.Services.AddScoped<ISoapService, SoapService>();
var app = builder.Build();
app.UseRouting();
app.UseEndpoints(endpoints =>
{
   _=endpoints.UseSoapEndpoint<ISoapService>("/Service.asmx",
new SoapEncoderOptions(), SoapSerializer.XmlSerializer, true);
});
//^ Case-insensitive paths
app.Run();
```



```
using SoapCore;
using SoapWebService.BusinessLogic;
var builder = WebApplication.CreateBuilder(args);
builder.Services.AddSoapCore();
builder.Services.AddScoped<ISoapService, SoapService>();
var app = builder.Build();
app.UseRouting();
app.UseEndpoints(endpoints =>
{
    _ = endpoints.UseSoapEndpoint<ISoapService>("/Service.asmx",new
      SoapEncoderOptions(), SoapSerializer.XmlSerializer, true);
    //^ Case insensitive paths
});
app.Run();
```

Add the ‘binary formatter’ override to the SoapWebService.proj project file, so that the APL64 CPC embedded workspace can be loaded into the project:

```
<EnableUnsafeBinaryFormatterSerialization>true</EnableUnsafeBinaryFormatterSerialization>
```



```
<Project Sdk="Microsoft.NET.Sdk.Web">

  <PropertyGroup>
    <TargetFramework>net8.0</TargetFramework>
    <Nullable>enable</Nullable>
    <ImplicitUsings>enable</ImplicitUsings>

      <EnableUnsafeBinaryFormatterSerialization>true</EnableUnsafeBinaryFormatterSerialization>

  </PropertyGroup>

  <ItemGroup>
    <None Remove="SoapWebService.http" />
  </ItemGroup>

  <ItemGroup>
    <PackageReference Include="APL64Programmer.SquareRootCpc" Version="1.0.0.1" />
    <PackageReference Include="SoapCore" Version="1.1.0.49" />
  </ItemGroup>

  <ItemGroup>
    <Folder Include="User Documentation\" />
  </ItemGroup>

</Project>
```

APL64 Cross-Platform Component (CPC)

For purposes of this document the APL64 CPC implementation files are contained in the 'APL64 Cross-Platform Component' folder of the SoapWebService solution.

Prepare an APL64 Workspace with an APL64 Public Function:

An APL64 programmer-defined, public function will support the square root algorithm in the web service. The 'APL64 Cross-Platform Component\CPC Source Folder\SquareRootCPC.ws64' workspace contains the required public function.

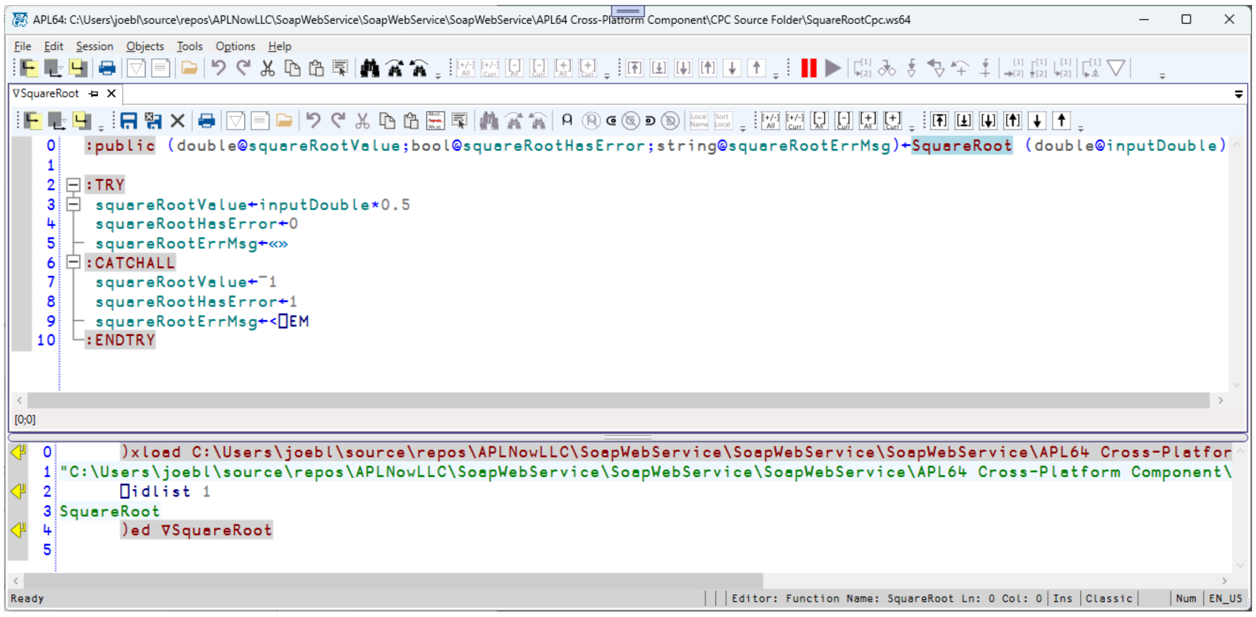
The APL64 public function, SquareRoot, has a single right argument, inputDouble, specified as a Double numeric value and three result values:

- squareRootValue, specified as a Double numeric value
- squareRootHasError, specifies as a Boolean value
- squareRootErrorMsg, specified as a String value

When the calculation of the squareRootValue occurs without an exception, the squareRootErrMsg is assigned an empty String, «». If the calculation of the squareRootValue results in an exception, the squareRootErrMsg is assigned the string value of the APL64 error message system variable, <⎕EM.

```
:public
(double@squareRootValue;bool@squareRootHasError;string@squareRootErrMsg)←SquareRoot
(double@inputDouble)

:TRY
 squareRootValue←inputDouble*0.5
 squareRootHasError←0
 squareRootErrMsg←«»
:CATCHALL
 squareRootValue←¯1
 squareRootHasError←1
 squareRootErrMsg←<⎕EM
:ENDTRY
```

In the APL64 developer version, load the 'APL64 Cross-Platform Component\CPC Source Folder\SquareRootCPC.ws64' workspace:

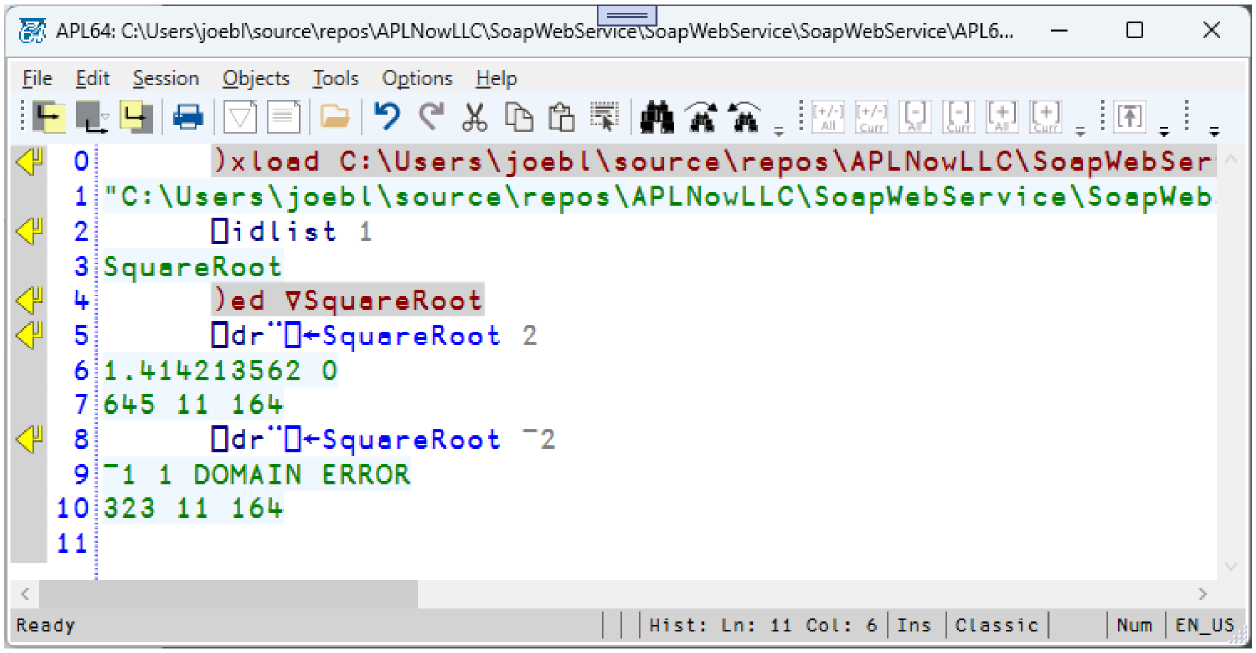

APL64: C:\Users\joebl\source\repos\APLNowLLC\SoapWebService\SoapWebService\SoapWebService\APL64 Cross-Platform Component\CPC Source Folder\SquareRootCpc.ws64

```
:public (double@squareRootValue;bool@squareRootHasError;string@squareRootErrMsg)←SquareRoot (double@inputDouble)

:TRY
 squareRootValue←inputDouble*0.5
 squareRootHasError←0
 squareRootErrMsg←«»
:CATCHALL
 squareRootValue←¯1
 squareRootHasError←1
 squareRootErrMsg←<⎕EM
:ENDTRY
```

```
      )xload C:\Users\joebl\source\repos\APLNowLLC\SoapWebService\SoapWebService\SoapWebService\APL64 Cross-Platfor
"C:\Users\joebl\source\repos\APLNowLLC\SoapWebService\SoapWebService\SoapWebService\APL64 Cross-Platform Component\
      ⎕idlist 1
SquareRoot
      )ed ∇SquareRoot
```

```
Ready    Editor: Function Name: SquareRoot Ln: 0 Col: 0 | Ins | Classic | Num | EN_US
```

Test the APL64 Public Function

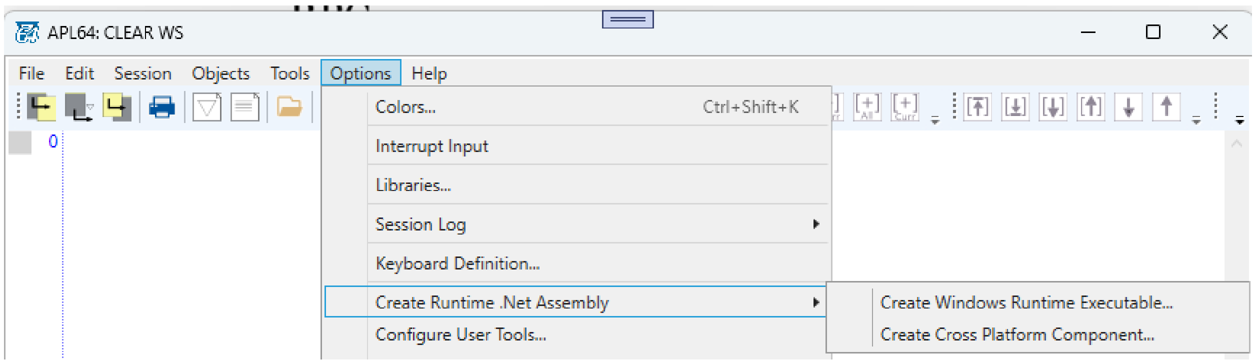

APL64: C:\Users\joebl\source\repos\APLNowLLC\SoapWebService\SoapWebService\SoapWebService\APL6...

```
      )xload C:\Users\joebl\source\repos\APLNowLLC\SoapWebSer
"C:\Users\joebl\source\repos\APLNowLLC\SoapWebService\SoapWeb
      ⎕idlist 1
SquareRoot
      )ed ∇SquareRoot
      ⎕dr¨⎕←SquareRoot 2
1.414213562 0
645 11 164
      ⎕dr¨⎕←SquareRoot ¯2
¯1 1 DOMAIN ERROR
323 11 164
```

```
Ready    Hist: Ln: 11 Col: 6 | Ins | Classic | Num | EN_US
```

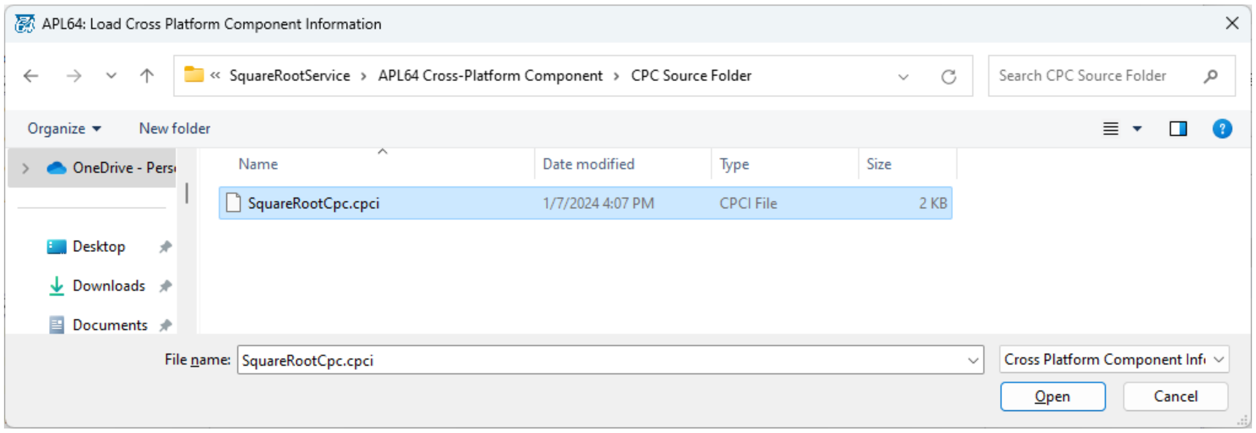
Use the APL64 Cross-platform Component Utility

Use Options | Create Runtime .Net Assembly | Create Cross-Platform Component utility in the APL64 Developer version:


APL64: CLEAR WS
File Edit Session Objects Tools Options Help
Colors... Ctrl+Shift+K
Interrupt Input
Libraries...
Session Log
Keyboard Definition...
Create Runtime .Net Assembly
Configure User Tools...
Create Windows Runtime Executable...
Create Cross Platform Component...

Use the ‘Load CPC Info’ button

Load the ‘APL64 Cross-Platform Component\CPC Source Folder\SquareRootCpc.cpci’ CPC configuration info file:


APL64: Load Cross Platform Component Information
« SquareRootService › APL64 Cross-Platform Component › CPC Source Folder
Search CPC Source Folder
Organize New folder
OneDrive - Pers
Desktop
Downloads
Documents
Open
Cancel

Name	Date modified	Type	Size
SquareRootCpc.cpci	1/7/2024 4:07 PM	CPCI File	2 KB

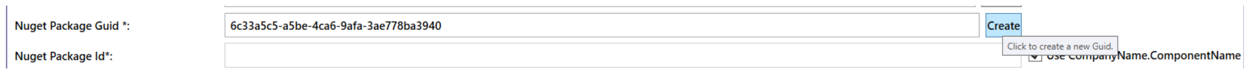
File name: SquareRootCpc.cpci
Cross Platform Component Info

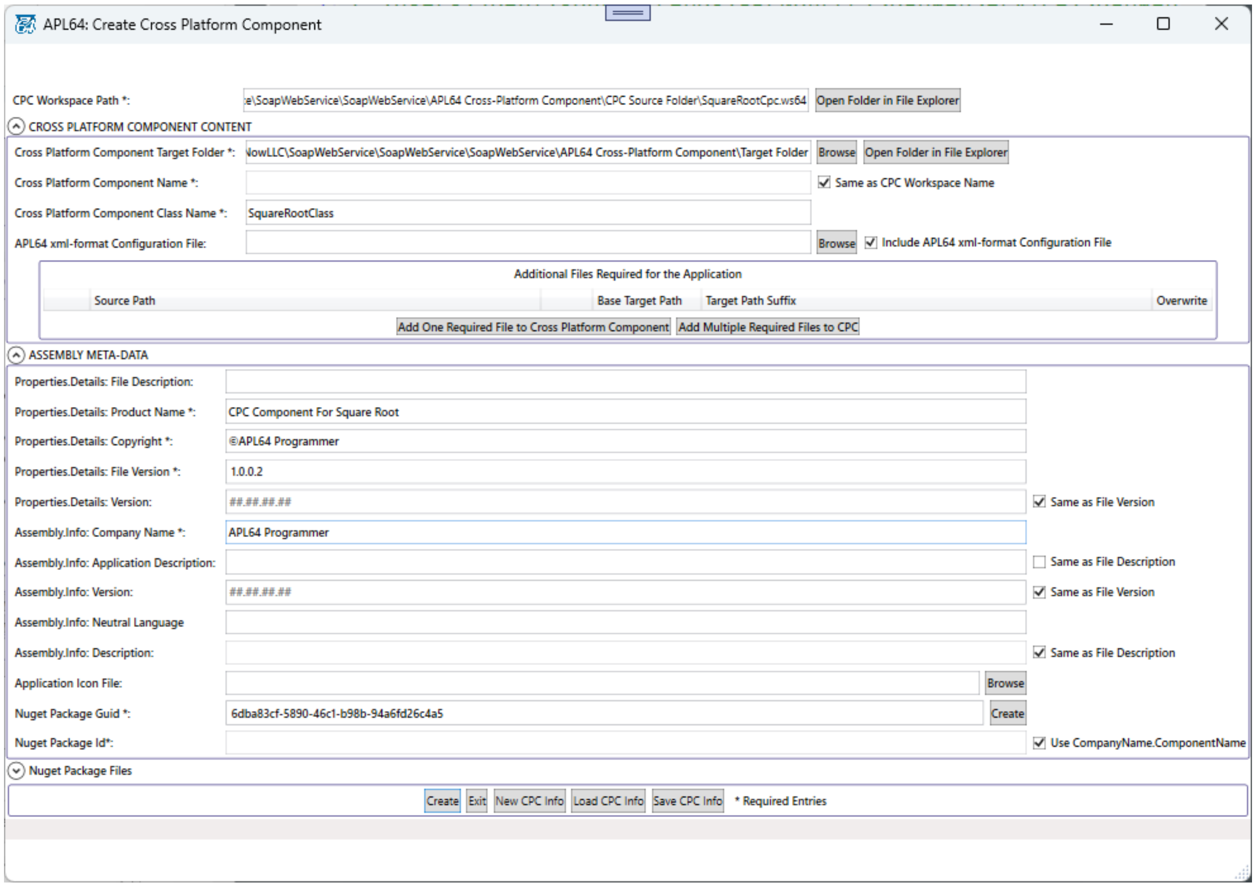
Update the ‘CPC Workspace Path’ and ‘Cross Platform Component Target Folder’ Entries

The entries should be modified as appropriate for the location of the SquareRoot service solution on the APL64 programmer’s workstation.

Update the Nuget Package Guid

Use the ‘Create’ button to the right of this entry in the dialog:


Nuget Package Guid *: 6c33a5c5-a5be-4ca6-9afa-3ae778ba3940
Nuget Package Id*:


APL64: Create Cross Platform Component

CPC Workspace Path *:	:e\SoapWebService\SoapWebService\APL64 Cross-Platform Component\CPC Source Folder\SquareRootCpc.ws64	Open Folder in File Explorer

CROSS PLATFORM COMPONENT CONTENT

Cross Platform Component Target Folder *:	NowLLC\SoapWebService\SoapWebService\SoapWebService\APL64 Cross-Platform Component\Target Folder	Browse / Open Folder in File Explorer
Cross Platform Component Name *:		✓ Same as CPC Workspace Name
Cross Platform Component Class Name *:	SquareRootClass	
APL64 xml-format Configuration File:		Browse ✓ Include APL64 xml-format Configuration File

Additional Files Required for the Application

Source Path	Base Target Path	Target Path Suffix	Overwrite

ASSEMBLY META-DATA

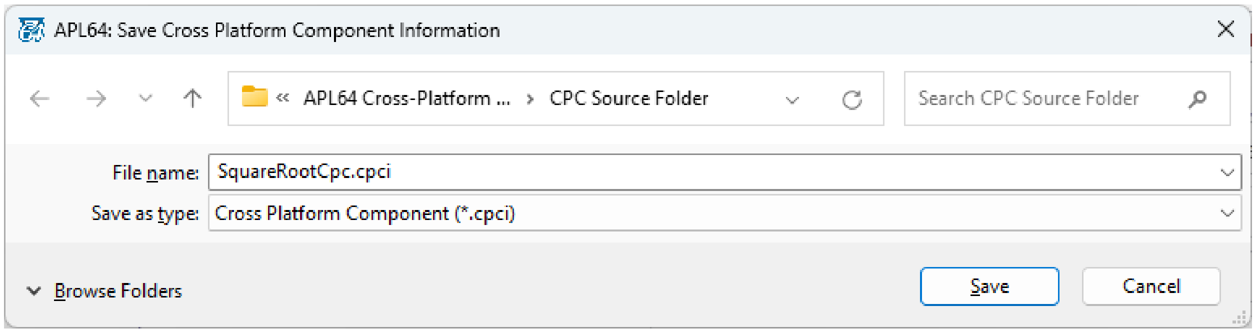
Properties.Details: File Description:		
Properties.Details: Product Name *:	CPC Component For Square Root	
Properties.Details: Copyright *:	©APL64 Programmer	
Properties.Details: File Version *:	1.0.0.2	
Properties.Details: Version:	##.##.##.##	✓ Same as File Version
Assembly.Info: Company Name *:	APL64 Programmer	
Assembly.Info: Application Description:		☐ Same as File Description
Assembly.Info: Version:	##.##.##.##	✓ Same as File Version
Assembly.Info: Neutral Language		
Assembly.Info: Description:		✓ Same as File Description
Application Icon File:		Browse
Nuget Package Guid *:	6dba83cf-5890-46c1-b98b-94a6fd26c4a5	Create
Nuget Package Id*:		✓ Use CompanyName.ComponentName

Nuget Package Files

* Required Entries

Add One Required File to Cross Platform Component
Add Multiple Required Files to CPC
Create
Exit
New CPC Info
Load CPC Info
Save CPC Info

Save the CPC Configuration File


APL64: Save Cross Platform Component Information

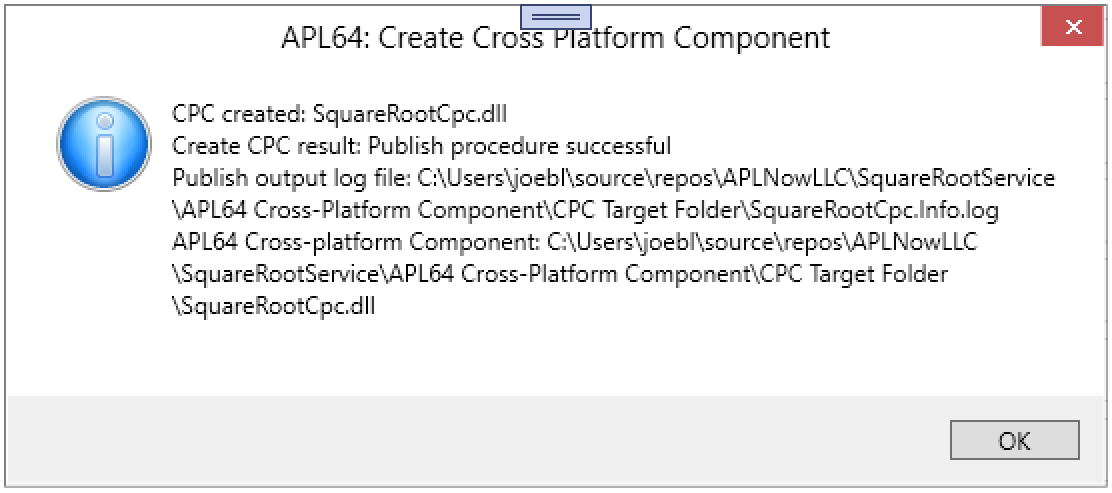
APL64 Cross-Platform ... › CPC Source Folder

File name: SquareRootCpc.cpci

Save as type: Cross Platform Component (*.cpci)

Create the Cross-platform Component

Click the Create button and observe the successful creation of the CPC:

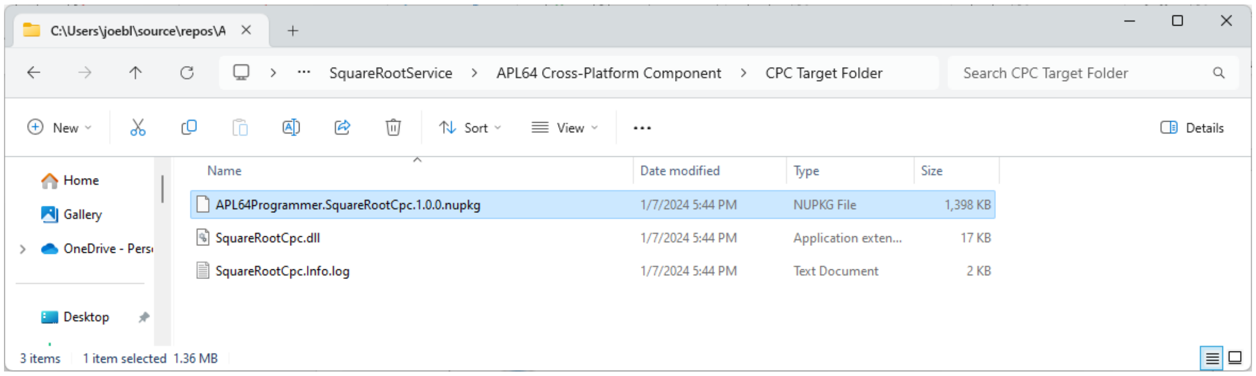


APL64: Create Cross Platform Component

```
CPC created: SquareRootCpc.dll
Create CPC result: Publish procedure successful
Publish output log file: C:\Users\joebl\source\repos\APLNowLLC\SquareRootService
\APL64 Cross-Platform Component\CPC Target Folder\SquareRootCpc.Info.log
APL64 Cross-platform Component: C:\Users\joebl\source\repos\APLNowLLC
\SquareRootService\APL64 Cross-Platform Component\CPC Target Folder
\SquareRootCpc.dll
```

Copy the APL64 CPC Nuget Package to a Local Nuget Publish Folder

Copy the ‘APL64Programmer.SquareRootCpc.1.0.0.nupkg’ Nuget package from the CPC Target Folder:



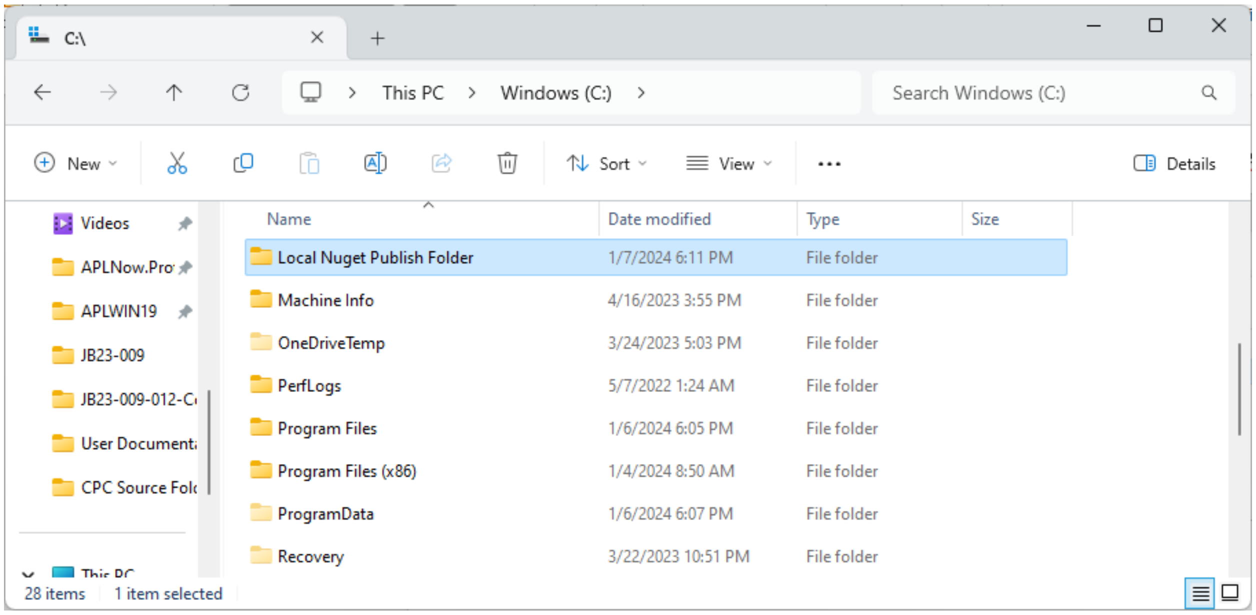
C:\Users\joebl\source\repos\A
SquareRootService > APL64 Cross-Platform Component > CPC Target Folder
Search CPC Target Folder
New
Sort
View
Details
Home
Gallery
OneDrive - Pers
Desktop

Name	Date modified	Type	Size
APL64Programmer.SquareRootCpc.1.0.0.nupkg	1/7/2024 5:44 PM	NUPKG File	1,398 KB
SquareRootCpc.dll	1/7/2024 5:44 PM	Application exten...	17 KB
SquareRootCpc.Info.log	1/7/2024 5:44 PM	Text Document	2 KB

3 items 1 item selected 1.36 MB

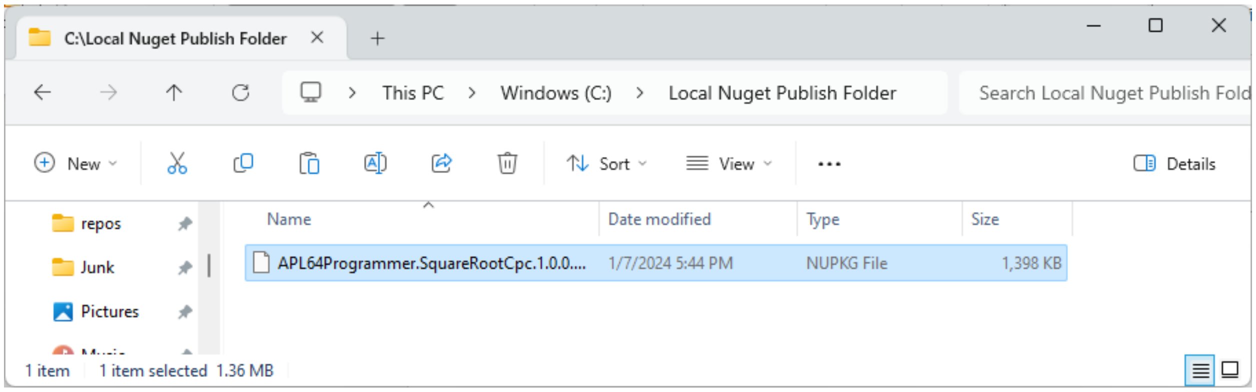
Create the ‘c:\ Local Nuget Publish Folder\’

This folder should not be in the SoapWebService solution folder.



Name	Date modified	Type	Size
Local Nuget Publish Folder	1/7/2024 6:11 PM	File folder	
Machine Info	4/16/2023 3:55 PM	File folder	
OneDriveTemp	3/24/2023 5:03 PM	File folder	
PerfLogs	5/7/2022 1:24 AM	File folder	
Program Files	1/6/2024 6:05 PM	File folder	
Program Files (x86)	1/4/2024 8:50 AM	File folder	
ProgramData	1/6/2024 6:07 PM	File folder	
Recovery	3/22/2023 10:51 PM	File folder	

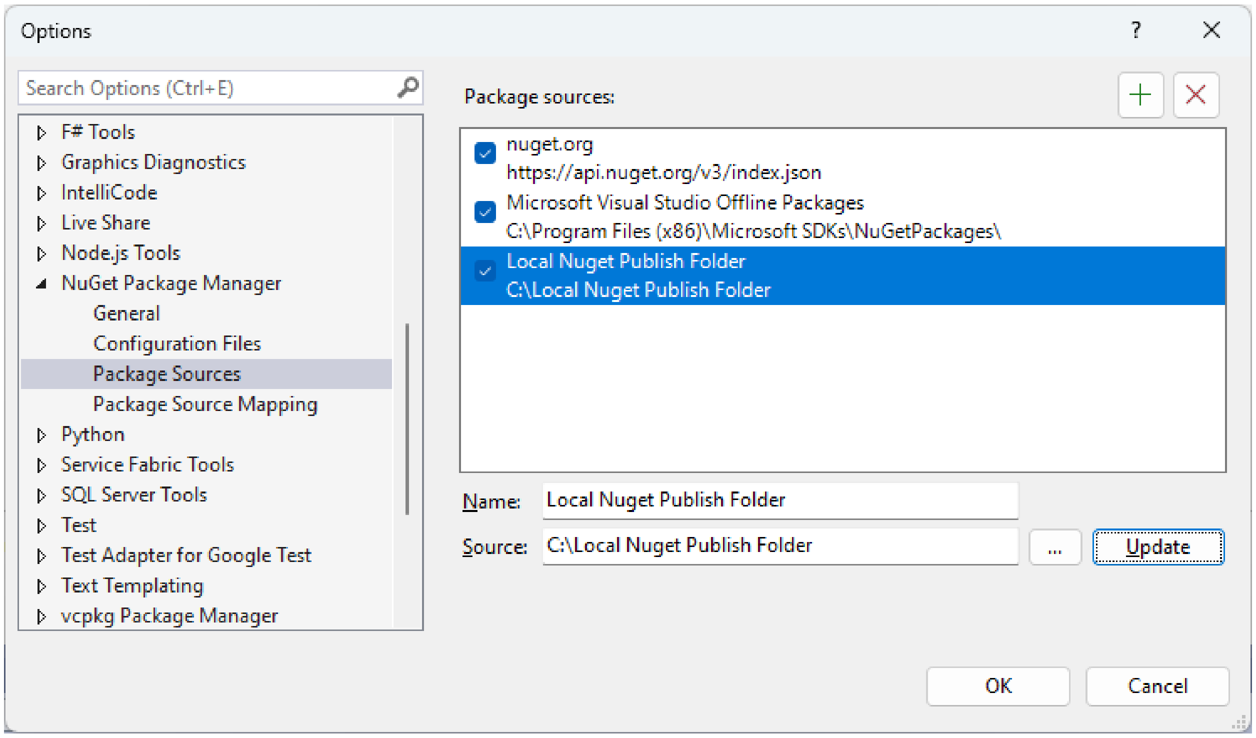
Paste the 'APL64Programmer.SquareRootCpc.1.0.0.nupkg' Nuget package to the 'c:\ Local Nuget Publish Folder\':


C:\Local Nuget Publish Folder
This PC > Windows (C:) > Local Nuget Publish Folder
Search Local Nuget Publish Fold
New
Sort
View
Details
repos
Junk
Pictures
Music
1 item | 1 item selected 1.36 MB

Name	Date modified	Type	Size
APL64Programmer.SquareRootCpc.1.0.0....	1/7/2024 5:44 PM	NUPKG File	1,398 KB

In VS2022 Recognize the Local Nuget Publish Folder

In Visual Studio use: Tools | Nuget Package Manager | Package Manager Settings | Package Sources | + (Add package source) to specify the Local Nuget Publish folder:



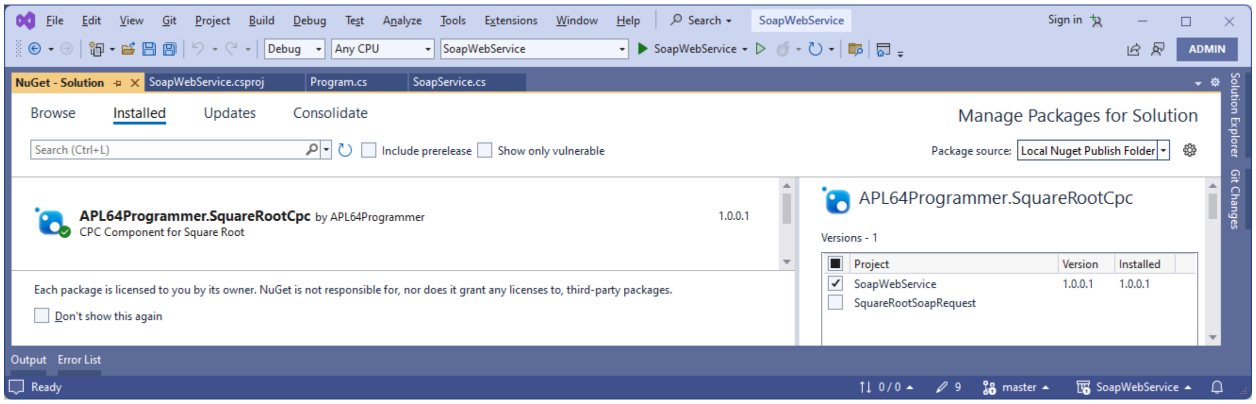
Package sources:

- [✓] nuget.org
  https://api.nuget.org/v3/index.json
- [✓] Microsoft Visual Studio Offline Packages
  C:\Program Files (x86)\Microsoft SDKs\NuGetPackages\
- [✓] Local Nuget Publish Folder
  C:\Local Nuget Publish Folder

Name: Local Nuget Publish Folder

Source: C:\Local Nuget Publish Folder

Install APL64 CPC Nuget package into SoapWebService



Package source: Local Nuget Publish Folder

APL64Programmer.SquareRootCpc by APL64Programmer 1.0.0.1
CPC Component for Square Root

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

[ ] Don't show this again

APL64Programmer.SquareRootCpc

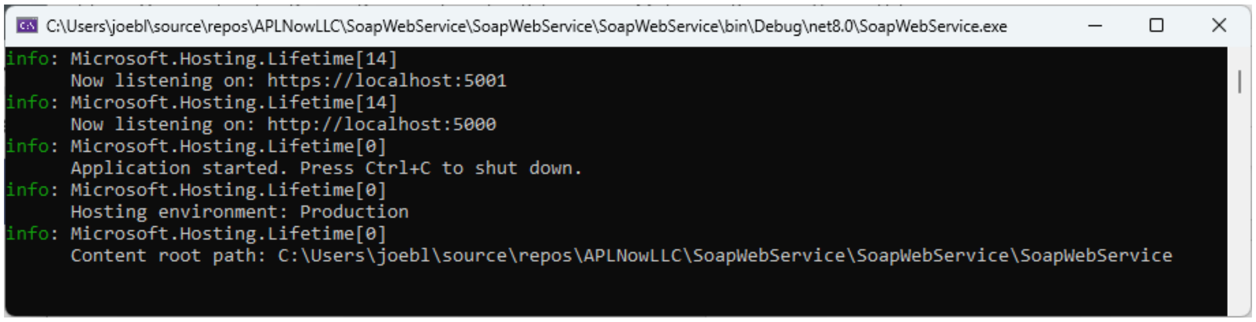
Versions - 1

	Project	Version	Installed
✓	SoapWebService	1.0.0.1	1.0.0.1
	SquareRootSoapRequest		

Start the SoapWebService

Debug the solution in VS2022 to start the web service

The URL and port for the http and https web transfer protocols are provided for use by a client of the web service.


C:\Users\joebl\source\repos\APLNowLLC\SoapWebService\SoapWebService\SoapWebService\bin\Debug\net8.0\SoapWebService.exe

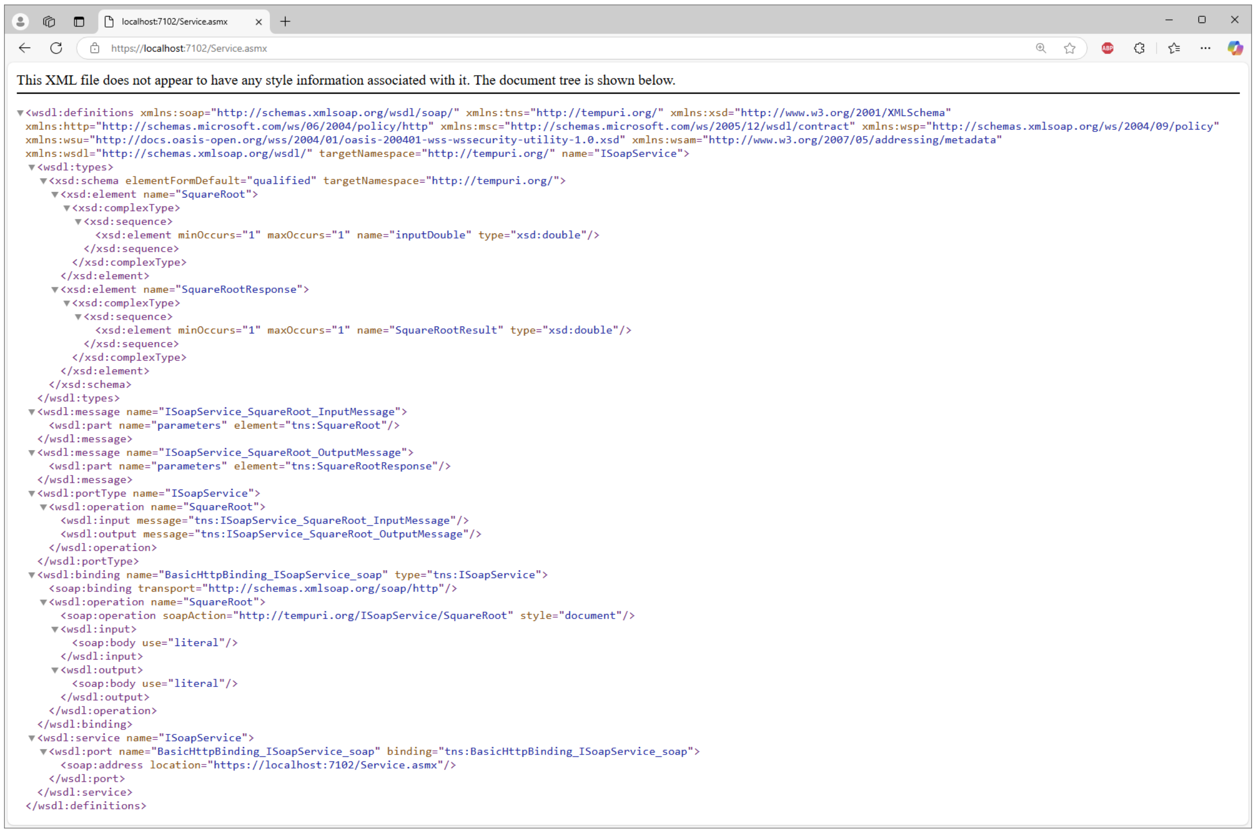
```
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:5001
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\joebl\source\repos\APLNowLLC\SoapWebService\SoapWebService\SoapWebService
```

The default web browser on the target workstation will open and display a 404 web exception, which is to be expected because the web service has no default web page.

In a web browser display the WSDL for the SOAP web service

The WSDL is a description of the web service which can be used by clients to access the web service. The WSDL uses the Web Service Description Language.

Using the web browser go to this url: https://localhost:7102/Service.asmx


localhost:7102/Service.asmx
https://localhost:7102/Service.asmx

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://tempuri.org/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:http="http://schemas.microsoft.com/ws/06/2004/policy/http" xmlns:msc="http://schemas.microsoft.com/ws/2005/12/wsdl/contract" xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
 xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
 xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://tempuri.org/" name="ISoapService">
  ▼<wsdl:types>
    ▼<xsd:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
      ▼<xsd:element name="SquareRoot">
        ▼<xsd:complexType>
          ▼<xsd:sequence>
              <xsd:element minOccurs="1" maxOccurs="1" name="inputDouble" type="xsd:double"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      ▼<xsd:element name="SquareRootResponse">
        ▼<xsd:complexType>
          ▼<xsd:sequence>
              <xsd:element minOccurs="1" maxOccurs="1" name="SquareRootResult" type="xsd:double"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:schema>
    </wsdl:types>
  ▼<wsdl:message name="ISoapService_SquareRoot_InputMessage">
      <wsdl:part name="parameters" element="tns:SquareRoot"/>
    </wsdl:message>
  ▼<wsdl:message name="ISoapService_SquareRoot_OutputMessage">
      <wsdl:part name="parameters" element="tns:SquareRootResponse"/>
    </wsdl:message>
  ▼<wsdl:portType name="ISoapService">
    ▼<wsdl:operation name="SquareRoot">
        <wsdl:input message="tns:ISoapService_SquareRoot_InputMessage"/>
        <wsdl:output message="tns:ISoapService_SquareRoot_OutputMessage"/>
      </wsdl:operation>
    </wsdl:portType>
  ▼<wsdl:binding name="BasicHttpBinding_ISoapService_soap" type="tns:ISoapService">
      <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    ▼<wsdl:operation name="SquareRoot">
        <soap:operation soapAction="http://tempuri.org/ISoapService/SquareRoot" style="document"/>
      ▼<wsdl:input>
          <soap:body use="literal"/>
        </wsdl:input>
      ▼<wsdl:output>
          <soap:body use="literal"/>
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
  ▼<wsdl:service name="ISoapService">
    ▼<wsdl:port name="BasicHttpBinding_ISoapService_soap" binding="tns:BasicHttpBinding_ISoapService_soap">
        <soap:address location="https://localhost:7102/Service.asmx"/>
      </wsdl:port>
    </wsdl:service>
  </wsdl:definitions>
```

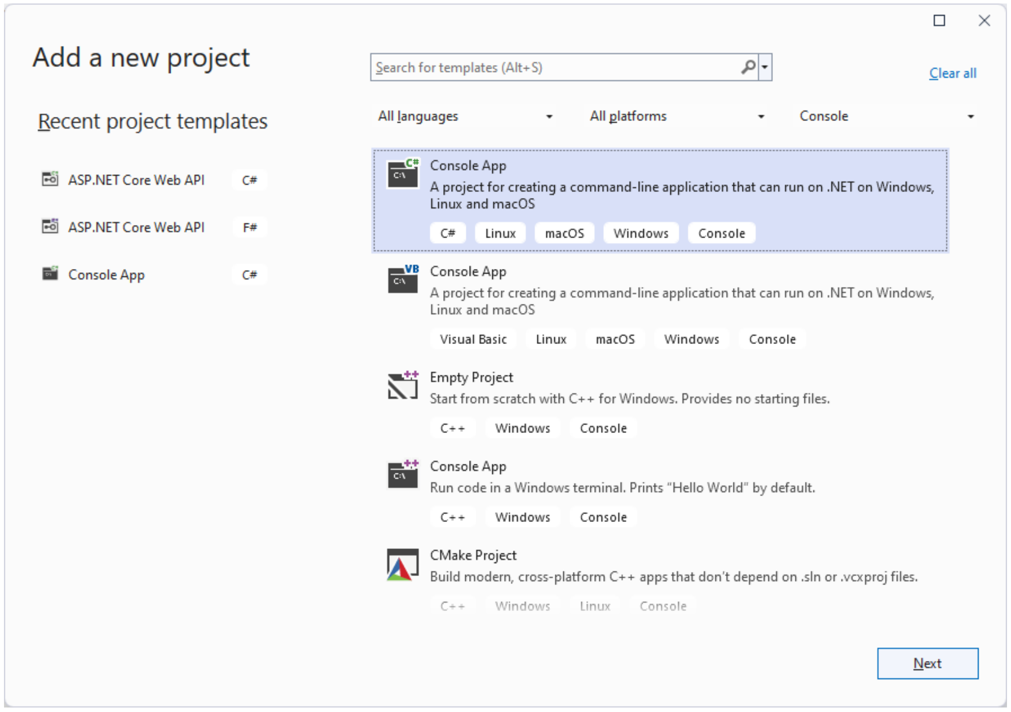
Browser Client to Access Access SoapWebService

In a production environment a client for the SoapWebService would be a server-to-server or end-use GUI application. For purposes of this example, a console project is used as the client.

Note: The SquareRootSoapRequest client described here is based on the Microsoft-deprecated 'WebRequest' technology, which has been replace by 'HttpClient' technology. The SquareRootSoapRequest client project was included in previous versions of this document.

See below in this document for an up-to-date 'SoapServiceHttpClient' project with the same functionality.

Add a new 'Console' project to the solution called SquareRootSoapRequest



Console App
A project for creating a command-line application that can run on .NET on Windows, Linux and macOS
C# Linux macOS Windows Console

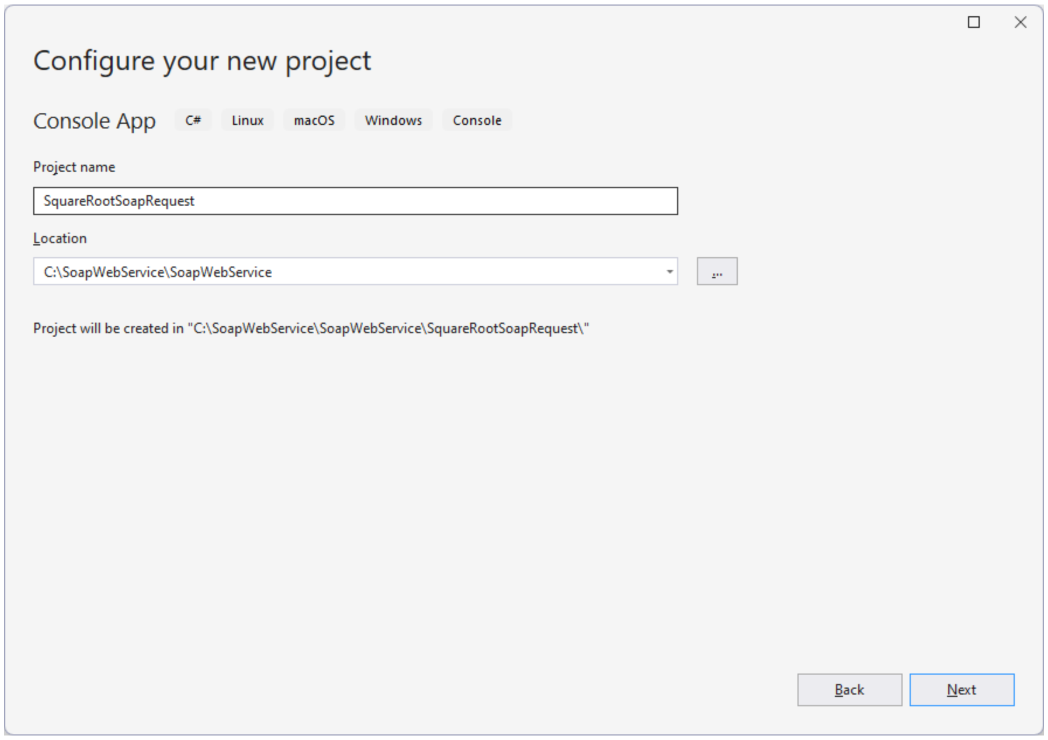
Console App
A project for creating a command-line application that can run on .NET on Windows, Linux and macOS
Visual Basic Linux macOS Windows Console

Empty Project
Start from scratch with C++ for Windows. Provides no starting files.
C++ Windows Console

Console App
Run code in a Windows terminal. Prints "Hello World" by default.
C++ Windows Console

CMake Project
Build modern, cross-platform C++ apps that don't depend on .sln or .vcxproj files.
C++ Windows Linux Console

Configure the Client project


Configure your new project

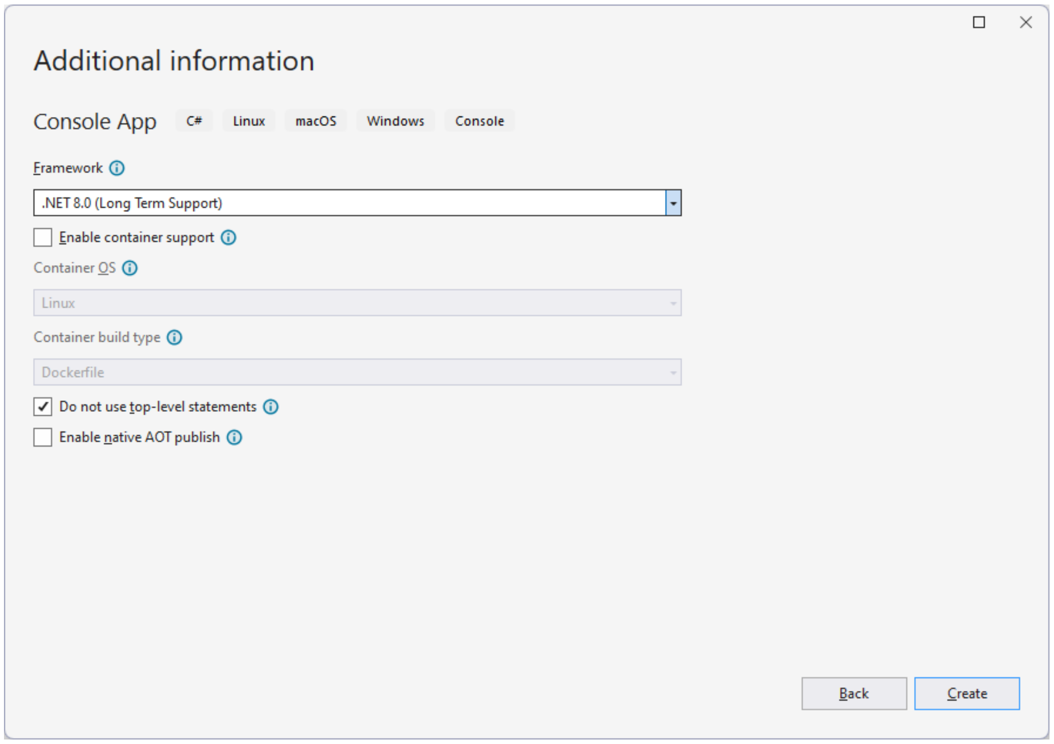
Console App C# Linux macOS Windows Console

Project name: SquareRootSoapRequest

Location: C:\SoapWebService\SoapWebService

Project will be created in "C:\SoapWebService\SoapWebService\SquareRootSoapRequest\"

Provide the Additional information for the Client project


Additional information

Console App C# Linux macOS Windows Console

Framework: .NET 8.0 (Long Term Support)

☐ Enable container support

Container OS: Linux

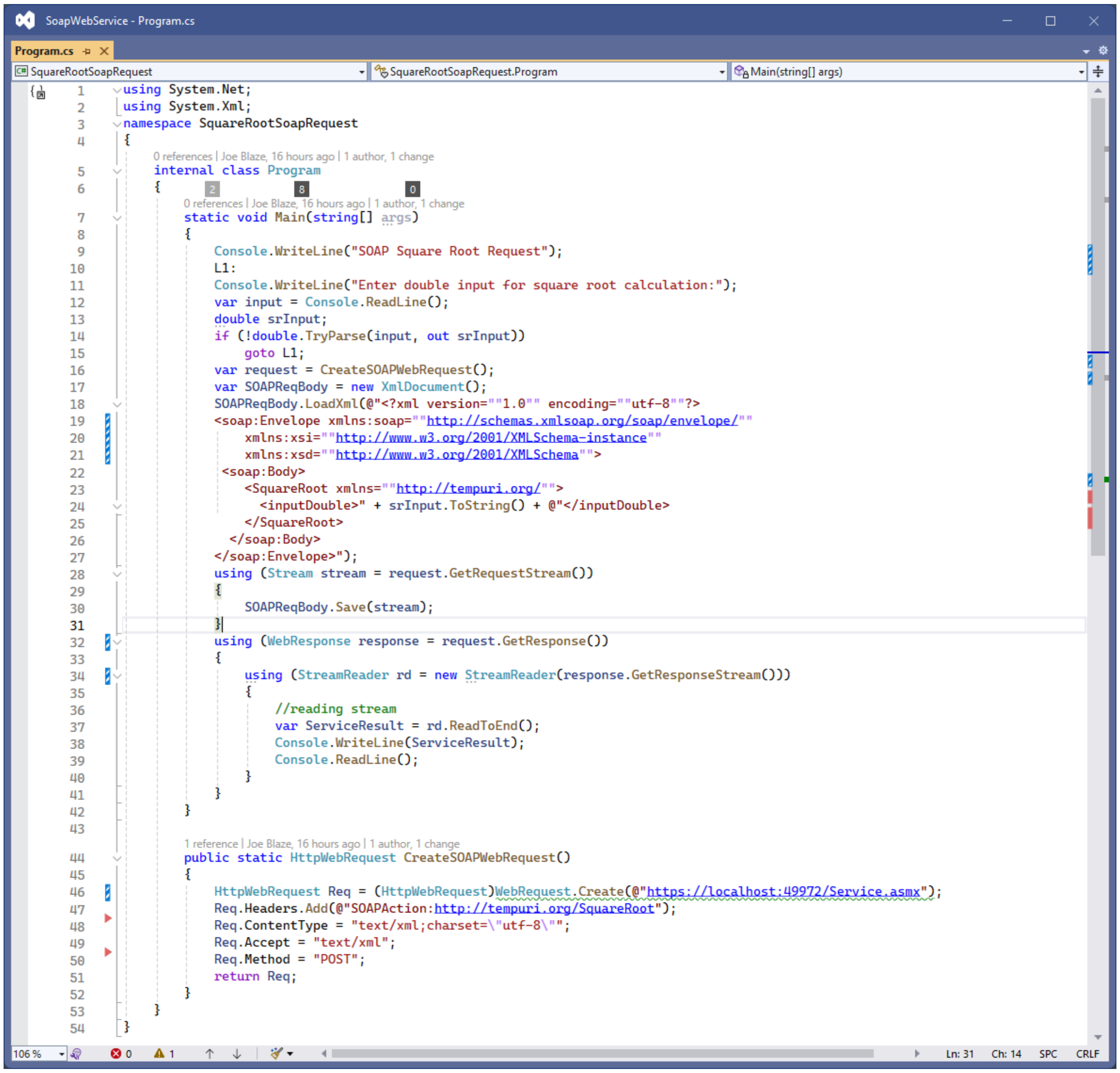
Container build type: Dockerfile

☑ Do not use top-level statements

☐ Enable native AOT publish

Replace Program.cs code file of Client project with this code:

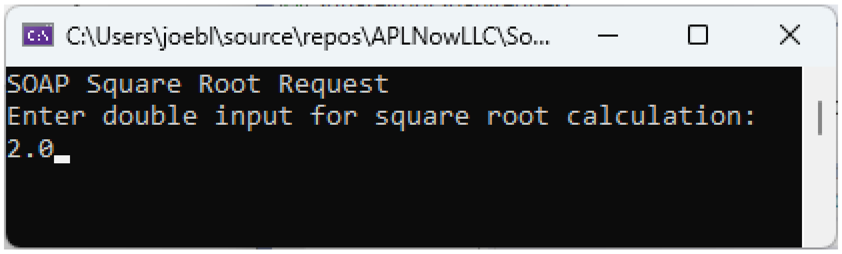
```
using System.Net;
using System.Xml;
namespace SquareRootSoapRequest
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("SOAP Square Root Request");
            L1:
            Console.WriteLine("Enter double input for square root calculation:");
            var input = Console.ReadLine();
            double srInput;
            if (!double.TryParse(input, out srInput))
                goto L1;
            var request = CreateSOAPWebRequest();
            var SOAPReqBody = new XmlDocument();
            SOAPReqBody.LoadXml(@"<?xml version=""1.0"" encoding=""utf-8""?>
            <soap:Envelope
xmlns:soap=""http://schemas.xmlsoap.org/soap/envelope/""
                xmlns:xsi=""http://www.w3.org/2001/XMLSchema-instance""
                xmlns:xsd=""http://www.w3.org/2001/XMLSchema"">
             <soap:Body>
                <SquareRoot xmlns=""http://tempuri.org/"">
                  <inputDouble>" + srInput.ToString() + @"</inputDouble>
                </SquareRoot>
              </soap:Body>
            </soap:Envelope>");
            using (Stream stream = request.GetRequestStream())
            {
                SOAPReqBody.Save(stream);
            }
            using (WebResponse response = request.GetResponse())
            {
                using (StreamReader rd = new
StreamReader(response.GetResponseStream()))
                {
                    //reading stream
                    var ServiceResult = rd.ReadToEnd();
                    Console.WriteLine(ServiceResult);
                    Console.ReadLine();
                }
            }
        }

        public static HttpWebRequest CreateSOAPWebRequest()
        {
            HttpWebRequest Req =
(HttpWebRequest)WebRequest.Create(@"https://localhost:7102/Service.asmx");
            Req.Headers.Add(@"SOAPAction:http://tempuri.org/SquareRoot");
            Req.ContentType = "text/xml;charset=\"utf-8\"";
            Req.Accept = "text/xml";
            Req.Method = "POST";
            return Req;
        }
    }
}
```



```
using System.Net;
using System.Xml;
namespace SquareRootSoapRequest
{
    0 references | Joe Blaze, 16 hours ago | 1 author, 1 change
    internal class Program
    {
        0 references | Joe Blaze, 16 hours ago | 1 author, 1 change
        static void Main(string[] args)
        {
            Console.WriteLine("SOAP Square Root Request");
            L1:
            Console.WriteLine("Enter double input for square root calculation:");
            var input = Console.ReadLine();
            double srInput;
            if (!double.TryParse(input, out srInput))
                goto L1;
            var request = CreateSOAPWebRequest();
            var SOAPReqBody = new XmlDocument();
            SOAPReqBody.LoadXml(@"<?xml version=""1.0"" encoding=""utf-8""?>
            <soap:Envelope xmlns:soap=""http://schemas.xmlsoap.org/soap/envelope/""
                xmlns:xsi=""http://www.w3.org/2001/XMLSchema-instance""
                xmlns:xsd=""http://www.w3.org/2001/XMLSchema"">
             <soap:Body>
                <SquareRoot xmlns=""http://tempuri.org/"">
                  <inputDouble>" + srInput.ToString() + @"</inputDouble>
                </SquareRoot>
              </soap:Body>
            </soap:Envelope>");
            using (Stream stream = request.GetRequestStream())
            {
                SOAPReqBody.Save(stream);
            }
            using (WebResponse response = request.GetResponse())
            {
                using (StreamReader rd = new StreamReader(response.GetResponseStream()))
                {
                    //reading stream
                    var ServiceResult = rd.ReadToEnd();
                    Console.WriteLine(ServiceResult);
                    Console.ReadLine();
                }
            }
        }

        1 reference | Joe Blaze, 16 hours ago | 1 author, 1 change
        public static HttpWebRequest CreateSOAPWebRequest()
        {
            HttpWebRequest Req = (HttpWebRequest)WebRequest.Create(@"https://localhost:49972/Service.asmx");
            Req.Headers.Add(@"SOAPAction:http://tempuri.org/SquareRoot");
            Req.ContentType = "text/xml;charset=\"utf-8\"";
            Req.Accept = "text/xml";
            Req.Method = "POST";
            return Req;
        }
    }
}
```

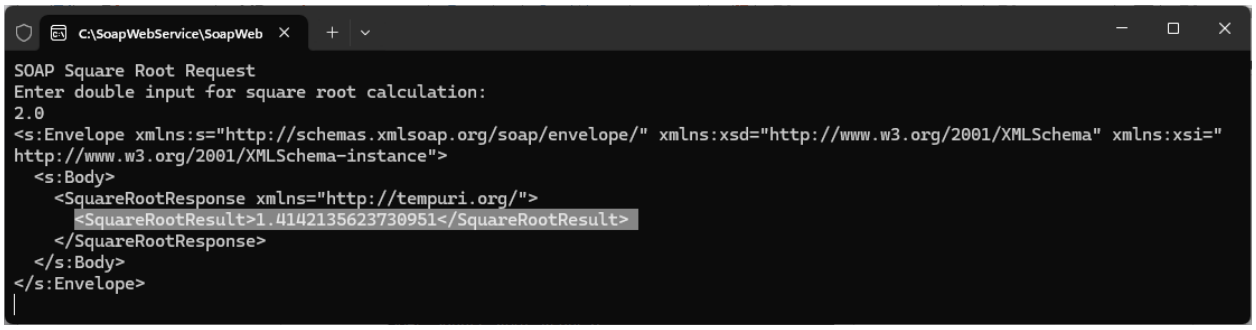
Make a Client Request to the SoapWebService

With the SoapWebService running, debug the SquareRootSoapRequest client project in Visual Studio, enter the desired double input value and click the Enter/Return key to calculate its square root:



```
SOAP Square Root Request
Enter double input for square root calculation:
2.0_
```

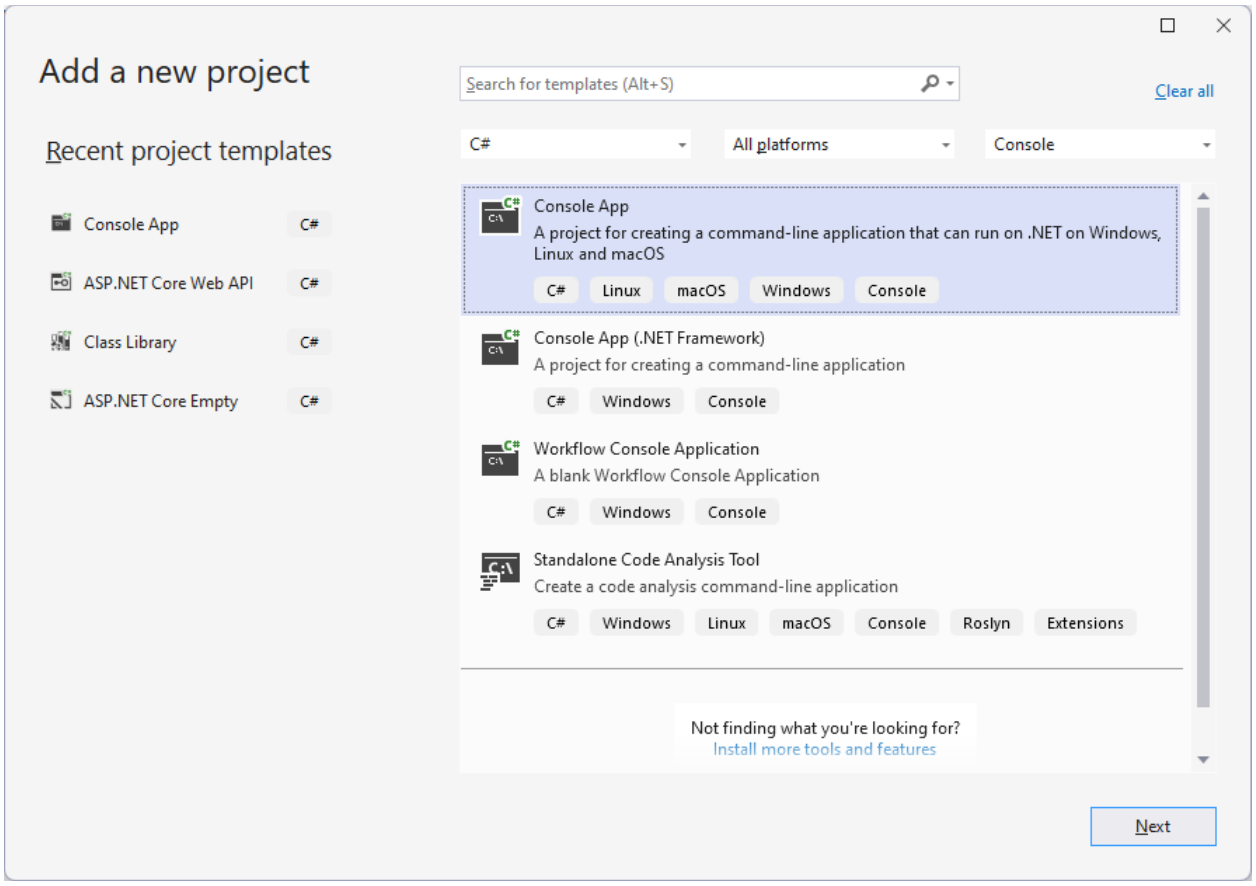
The resulting XML-format response, enclosed in a SOAP envelope in xml-format, which can be parsed by the client application as necessary:



```
SOAP Square Root Request
Enter double input for square root calculation:
2.0
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance">
  <s:Body>
    <SquareRootResponse xmlns="http://tempuri.org/">
      <SquareRootResult>1.4142135623730951</SquareRootResult>
    </SquareRootResponse>
  </s:Body>
</s:Envelope>
```

Request Square Root using HttpClient

Add the new ‘SoapServiceHttpClient’ console project to the ‘SoapWebService’ solution.



Add a new project

Recent project templates

- Console App C#
- ASP.NET Core Web API C#
- Class Library C#
- ASP.NET Core Empty C#

Search for templates (Alt+S)   Clear all

C#   All platforms   Console

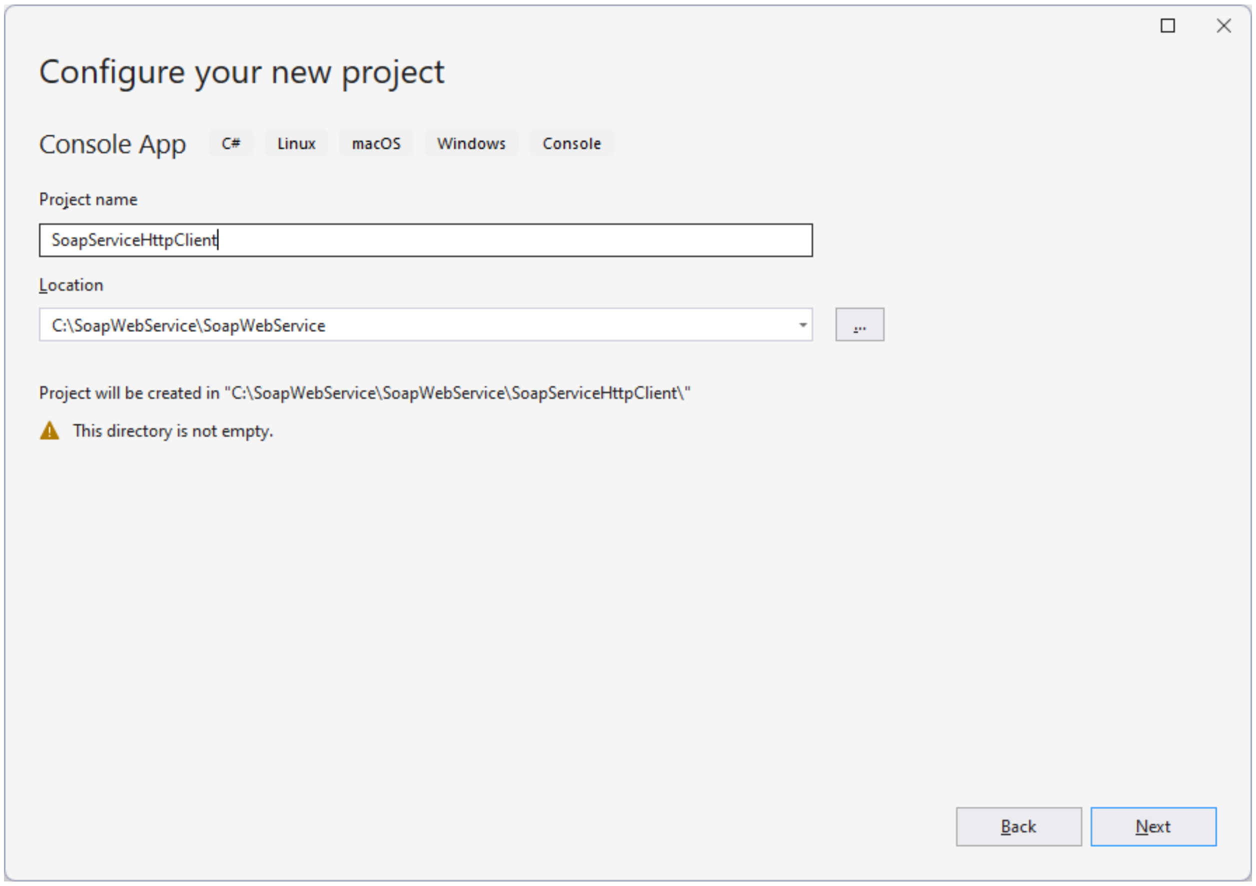
Console App
A project for creating a command-line application that can run on .NET on Windows, Linux and macOS
C# Linux macOS Windows Console

Console App (.NET Framework)
A project for creating a command-line application
C# Windows Console

Workflow Console Application
A blank Workflow Console Application
C# Windows Console

Standalone Code Analysis Tool
Create a code analysis command-line application
C# Windows Linux macOS Console Roslyn Extensions

Not finding what you're looking for?
Install more tools and features


Configure your new project

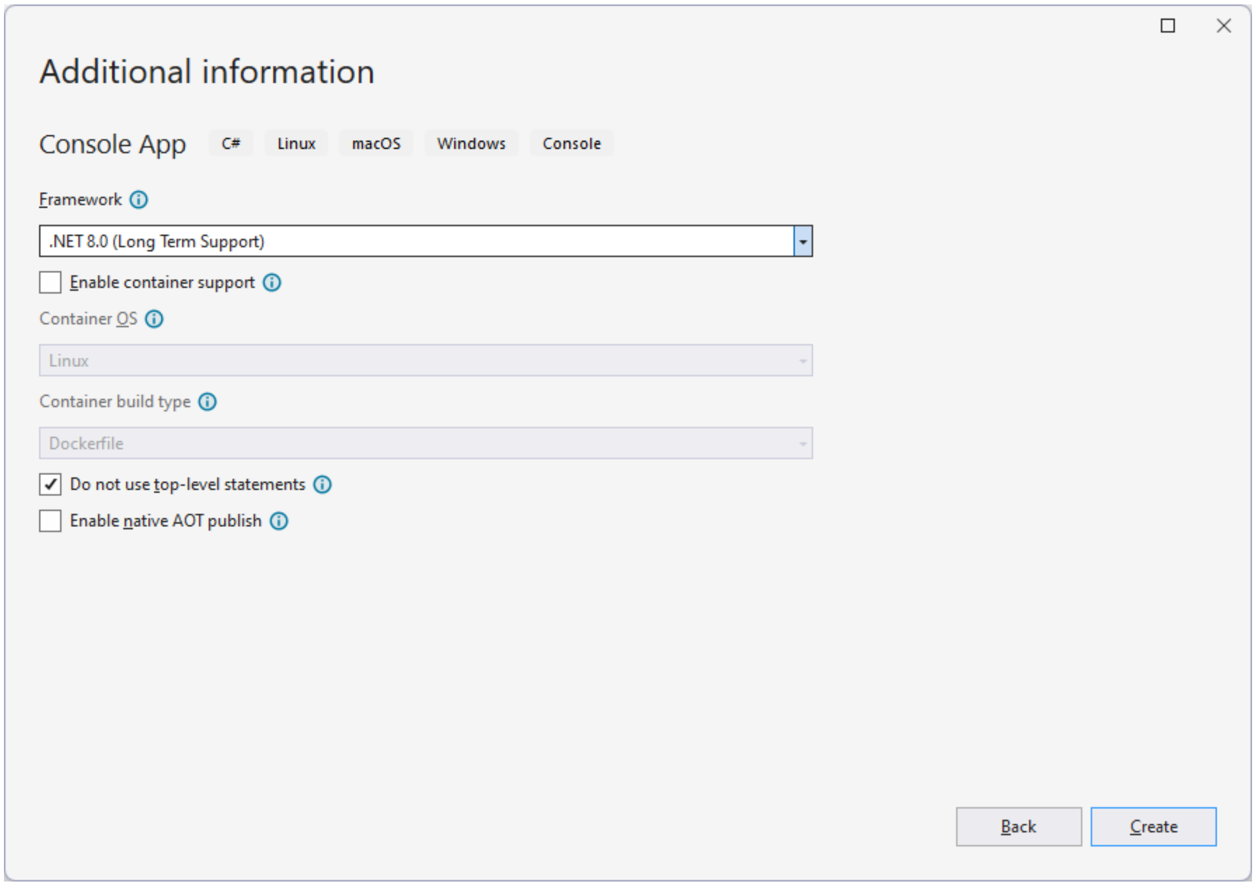
Console App

Project name: SoapServiceHttpClient

Location: C:\SoapWebService\SoapWebService

Project will be created in "C:\SoapWebService\SoapWebService\SoapServiceHttpClient\"

This directory is not empty.


Additional information

Console App C# Linux macOS Windows Console

Framework: .NET 8.0 (Long Term Support)

☐ Enable container support

Container OS: Linux

Container build type: Dockerfile

☑ Do not use top-level statements

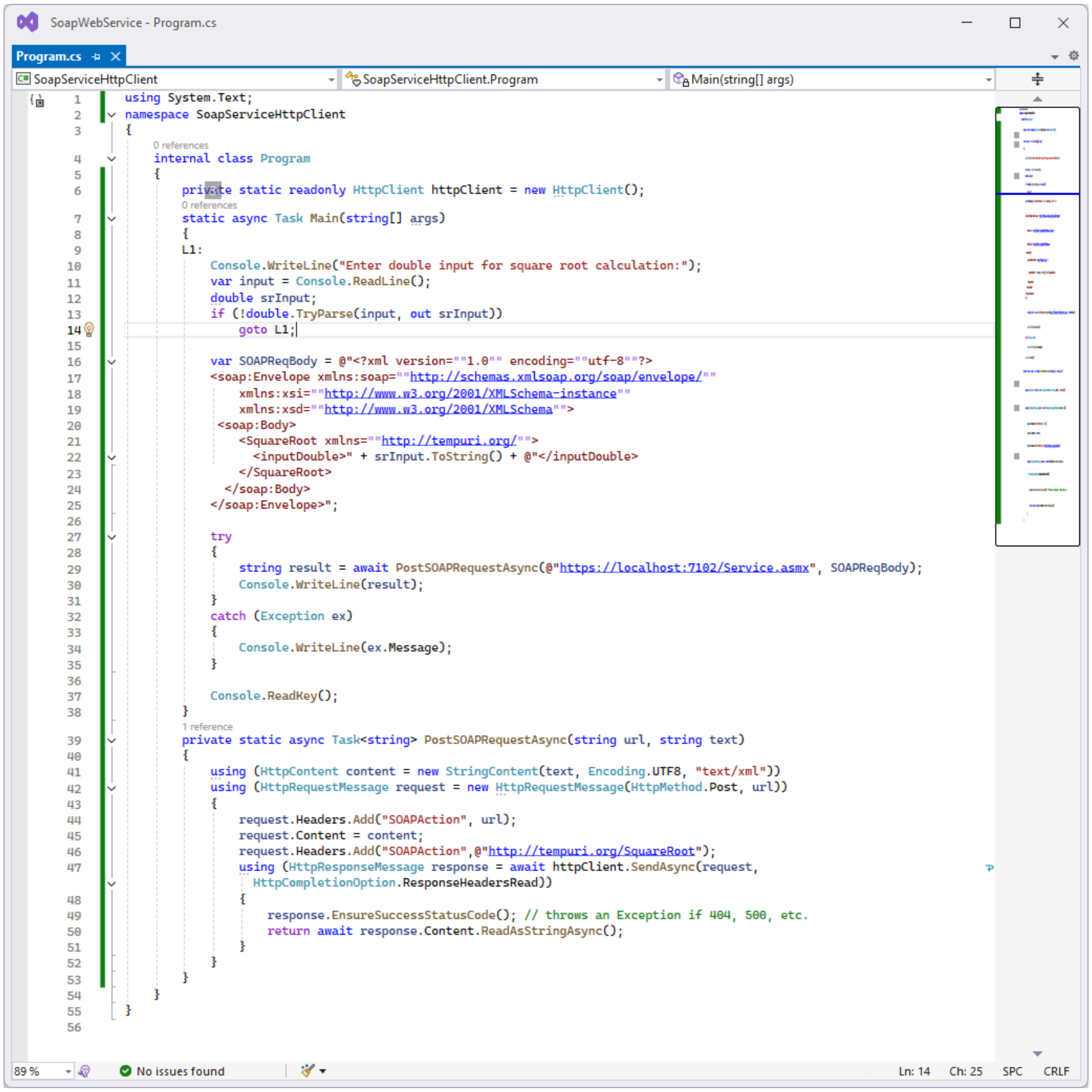
☐ Enable native AOT publish

Back Create

Replace the content of the Program.cs code file in the SoapServiceHttpClient project with this code:

```
using System.Text;
namespace SoapServiceHttpClient
{
  internal class Program
  {
    private static readonly HttpClient httpClient = new HttpClient();
    static async Task Main(string[] args)
    {
    L1:
      Console.WriteLine("Enter double input for square root calculation:");
      var input = Console.ReadLine();
      double srInput;
      if (!double.TryParse(input, out srInput))
        goto L1;

      var SOAPReqBody = @"<?xml version=""1.0"" encoding=""utf-8""?>
      <soap:Envelope xmlns:soap=""http://schemas.xmlsoap.org/soap/envelope/""
        xmlns:xsi=""http://www.w3.org/2001/XMLSchema-instance""
```

```
            xmlns:xsd=""http://www.w3.org/2001/XMLSchema"">
          <soap:Body>
            <SquareRoot xmlns=""http://tempuri.org/"">
              <inputDouble>" + srInput.ToString() + @"</inputDouble>
            </SquareRoot>
          </soap:Body>
        </soap:Envelope>";

        try
        {
            string result = await PostSOAPRequestAsync(@"https://localhost:7102/Service.asmx",
SOAPReqBody);
            Console.WriteLine(result);
        }
        catch (Exception ex)
        {
            Console.WriteLine(ex.Message);
        }

        Console.ReadKey();
    }
    private static async Task<string> PostSOAPRequestAsync(string url, string text)
    {
        using (HttpContent content = new StringContent(text, Encoding.UTF8, "text/xml"))
        using (HttpRequestMessage request = new HttpRequestMessage(HttpMethod.Post, url))
        {
            request.Headers.Add("SOAPAction", url);
            request.Content = content;
            request.Headers.Add("SOAPAction",@"http://tempuri.org/SquareRoot");
            using (HttpResponseMessage response = await httpClient.SendAsync(request,
HttpCompletionOption.ResponseHeadersRead))
            {
                response.EnsureSuccessStatusCode(); // throws an Exception if 404, 500, etc.
                return await response.Content.ReadAsStringAsync();
            }
        }
    }
  }
}
```

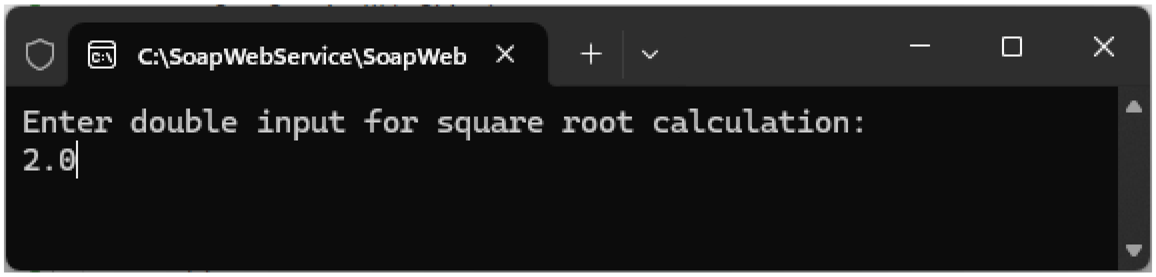


```csharp
using System.Text;
namespace SoapServiceHttpClient
{
    internal class Program
    {
        private static readonly HttpClient httpClient = new HttpClient();
        static async Task Main(string[] args)
        {
        L1:
            Console.WriteLine("Enter double input for square root calculation:");
            var input = Console.ReadLine();
            double srInput;
            if (!double.TryParse(input, out srInput))
                goto L1;

            var SOAPReqBody = @"<?xml version=""1.0"" encoding=""utf-8""?>
            <soap:Envelope xmlns:soap=""http://schemas.xmlsoap.org/soap/envelope/""
                xmlns:xsi=""http://www.w3.org/2001/XMLSchema-instance""
                xmlns:xsd=""http://www.w3.org/2001/XMLSchema"">
             <soap:Body>
                <SquareRoot xmlns=""http://tempuri.org/"">
                  <inputDouble>" + srInput.ToString() + @"</inputDouble>
                </SquareRoot>
              </soap:Body>
            </soap:Envelope>";

            try
            {
                string result = await PostSOAPRequestAsync(@"https://localhost:7102/Service.asmx", SOAPReqBody);
                Console.WriteLine(result);
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.Message);
            }

            Console.ReadKey();
        }
        private static async Task<string> PostSOAPRequestAsync(string url, string text)
        {
            using (HttpContent content = new StringContent(text, Encoding.UTF8, "text/xml"))
            using (HttpRequestMessage request = new HttpRequestMessage(HttpMethod.Post, url))
            {
                request.Headers.Add("SOAPAction", url);
                request.Content = content;
                request.Headers.Add("SOAPAction",@"http://tempuri.org/SquareRoot");
                using (HttpResponseMessage response = await httpClient.SendAsync(request,
                  HttpCompletionOption.ResponseHeadersRead))
                {
                    response.EnsureSuccessStatusCode(); // throws an Exception if 404, 500, etc.
                    return await response.Content.ReadAsStringAsync();
                }
            }
        }
    }
}
```

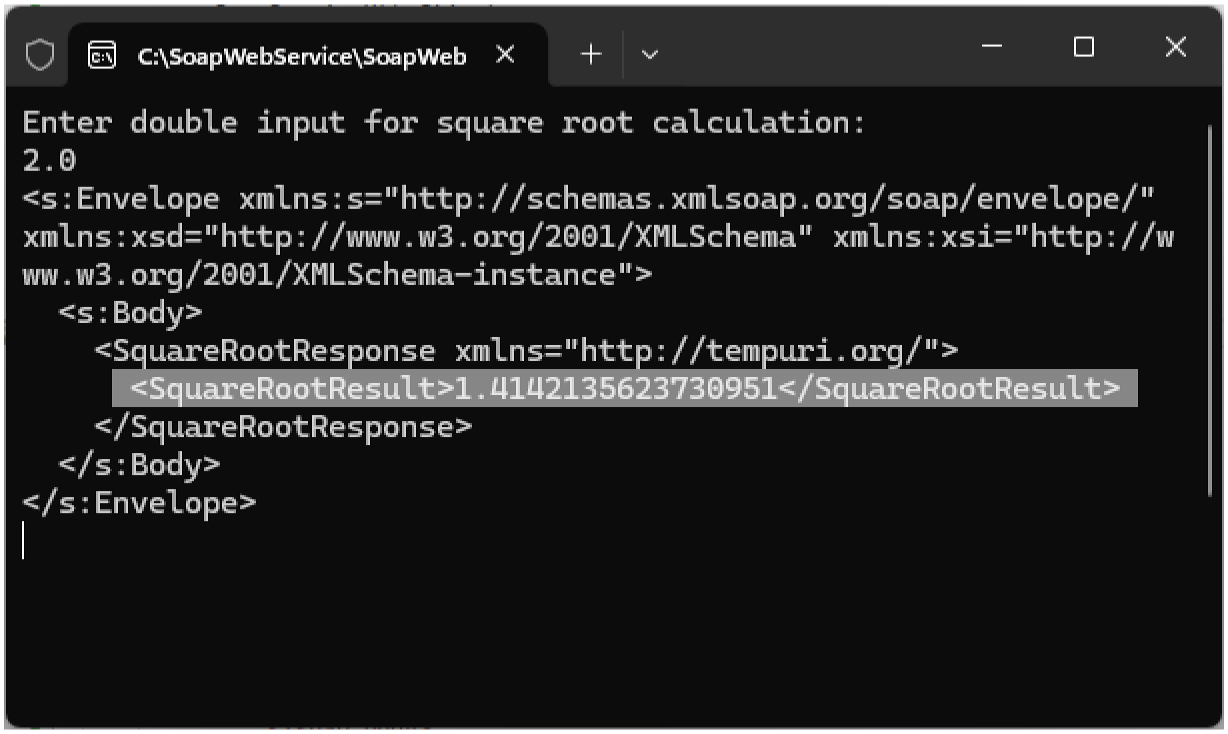
The source code in Program.cs:

- Requests user input of a double numeric value
- Creates xml-format text containing a SOAP envelope
- Adds xml-format SOAP body text with the request for a square root calculation
- Sends the SOAP envelope to the SoapWebService as a POST request
- Displays the result for the user

With the SoapWebService running, debug the SoapServiceHttpClient project, enter the desired double numeric input and click Enter/Return to see the SoapServiceHttpClient in action:



```
Enter double input for square root calculation:
2.0
```



```
Enter double input for square root calculation:
2.0
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://w
ww.w3.org/2001/XMLSchema-instance">
  <s:Body>
    <SquareRootResponse xmlns="http://tempuri.org/">
      <SquareRootResult>1.4142135623730951</SquareRootResult>
    </SquareRootResponse>
  </s:Body>
</s:Envelope>
```

More Information

For technical information for APL64 subscribers contact support@apl2000.com

For customized consulting or licensing information contact sales@apl2000