# Use an APL64 CPC in Excel

## Contents



## Overview

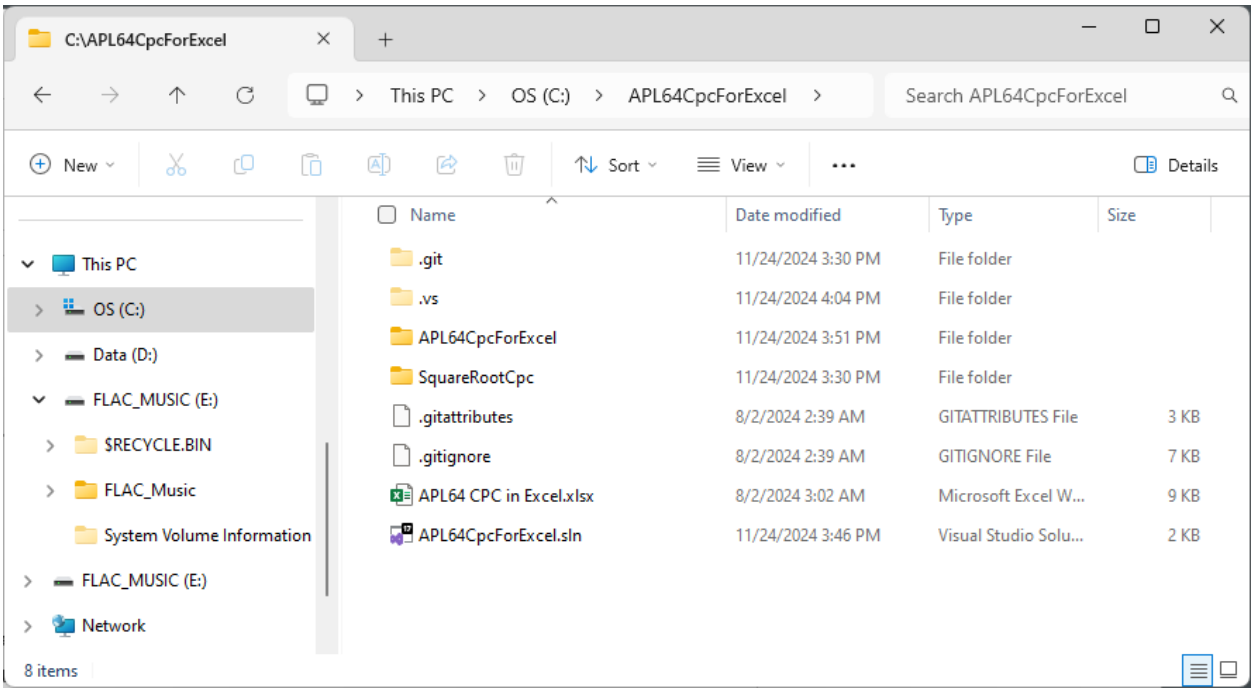
An APL64 cross-platform component (CPC) can support application-specific Excel functions, which are transparently available for use in Excel worksheet cells or macros.

To experience this example, Microsoft Excel must be installed on the target workstation.

## This example may be downloaded

Download source: http://apl2000.com/APL64/UserDocumentation/APL64_CPC_IN_EXCEL.zip

After downloading the zip-format file, extract it to 'c:\' to create the folder 'c:\APL64CpcForExcel\' containing these sub-folders:




If the downloaded example is used, the APL64 CPC must be created following the process described in this document.

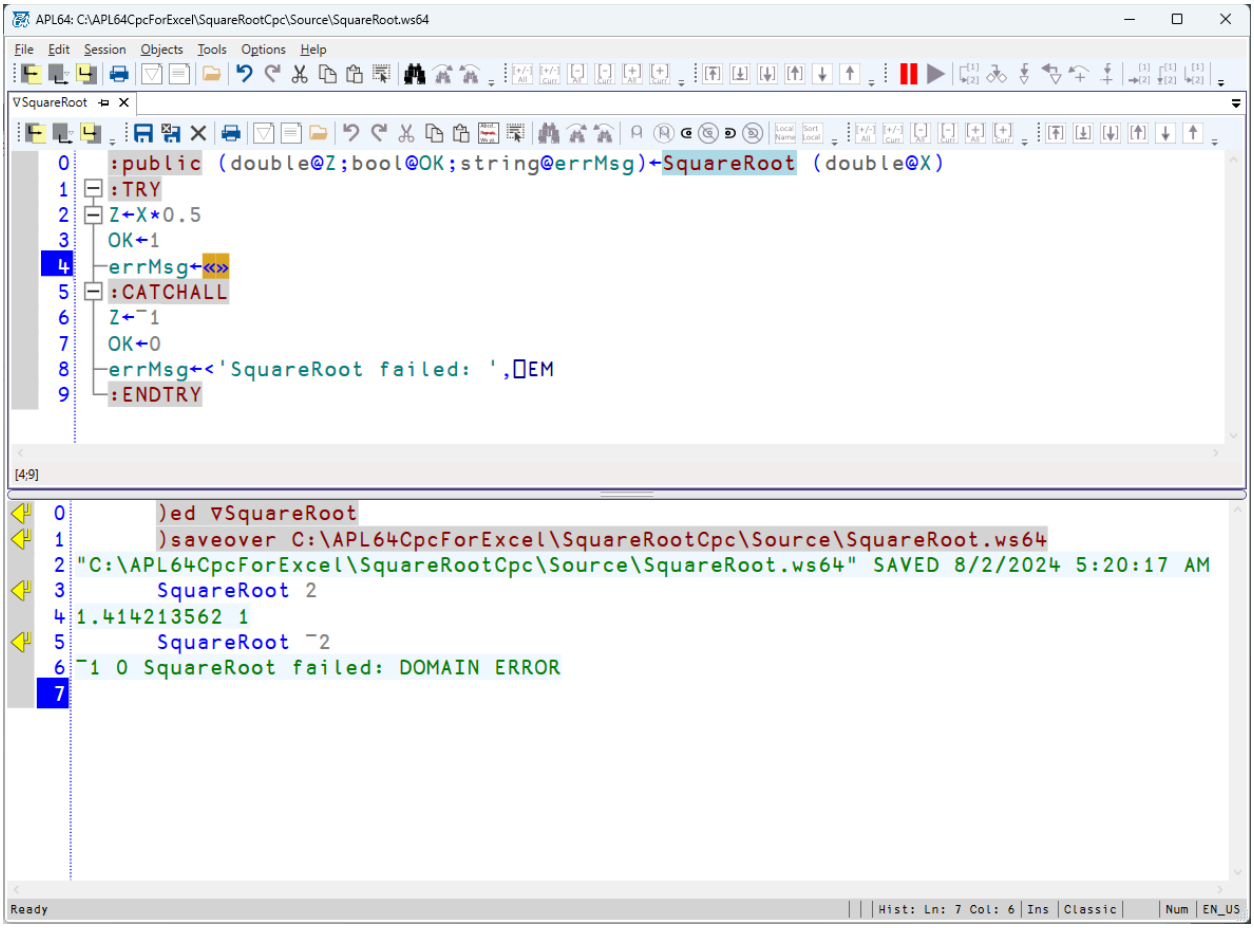
## Create an APL64 CPC

Start an instance of the APL64 developer version.

An APL64 CPC can expose any number of APL64 programmer-defined, public functions. An APL64 CPC workspace can also include APL64 programmer-defined, non-public functions necessary to support the application system's algorithms. The non-public functions are not available in Excel.

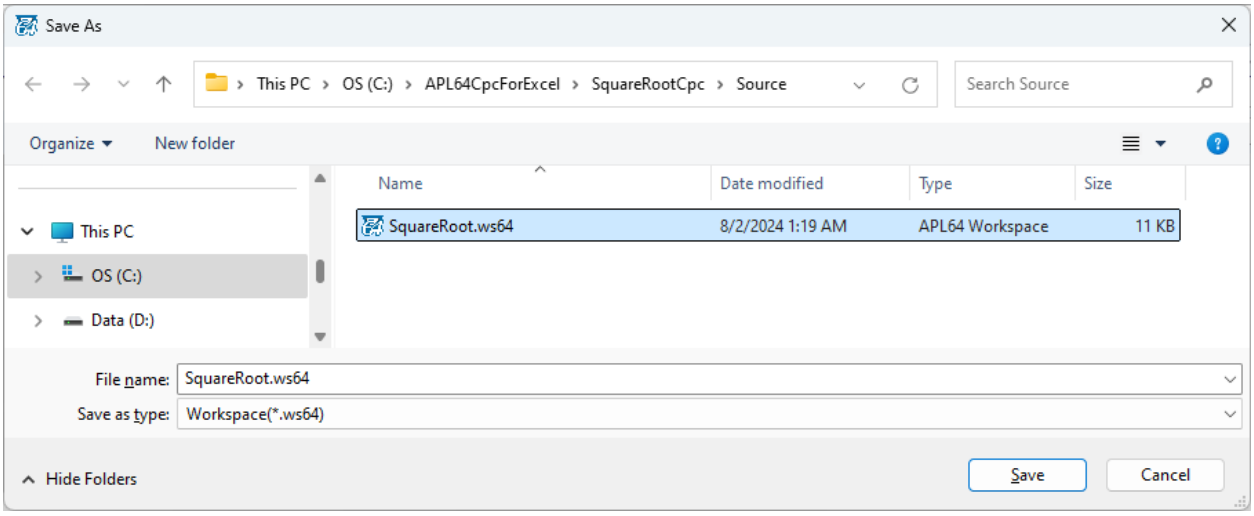
### Create the APL64 programmer-defined, public function SquareRoot:

```
:public (double@Z;bool@OK;string@errMsg)←SquareRoot (double@X)
:TRY
Z←X*0.5
OK←1
errMsg←«»
:CATCHALL
Z←¯1
OK←0
errMsg←<'SquareRoot failed: ',⎕EM
:ENDTRY
```

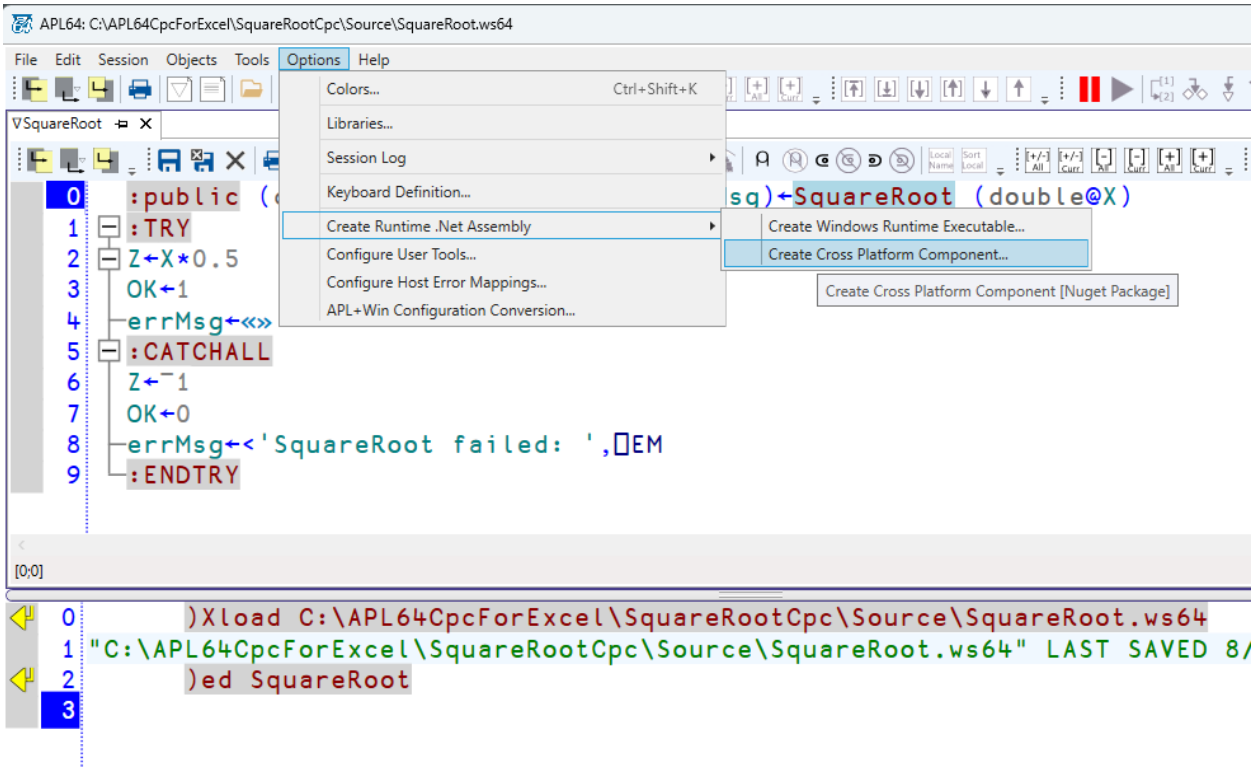


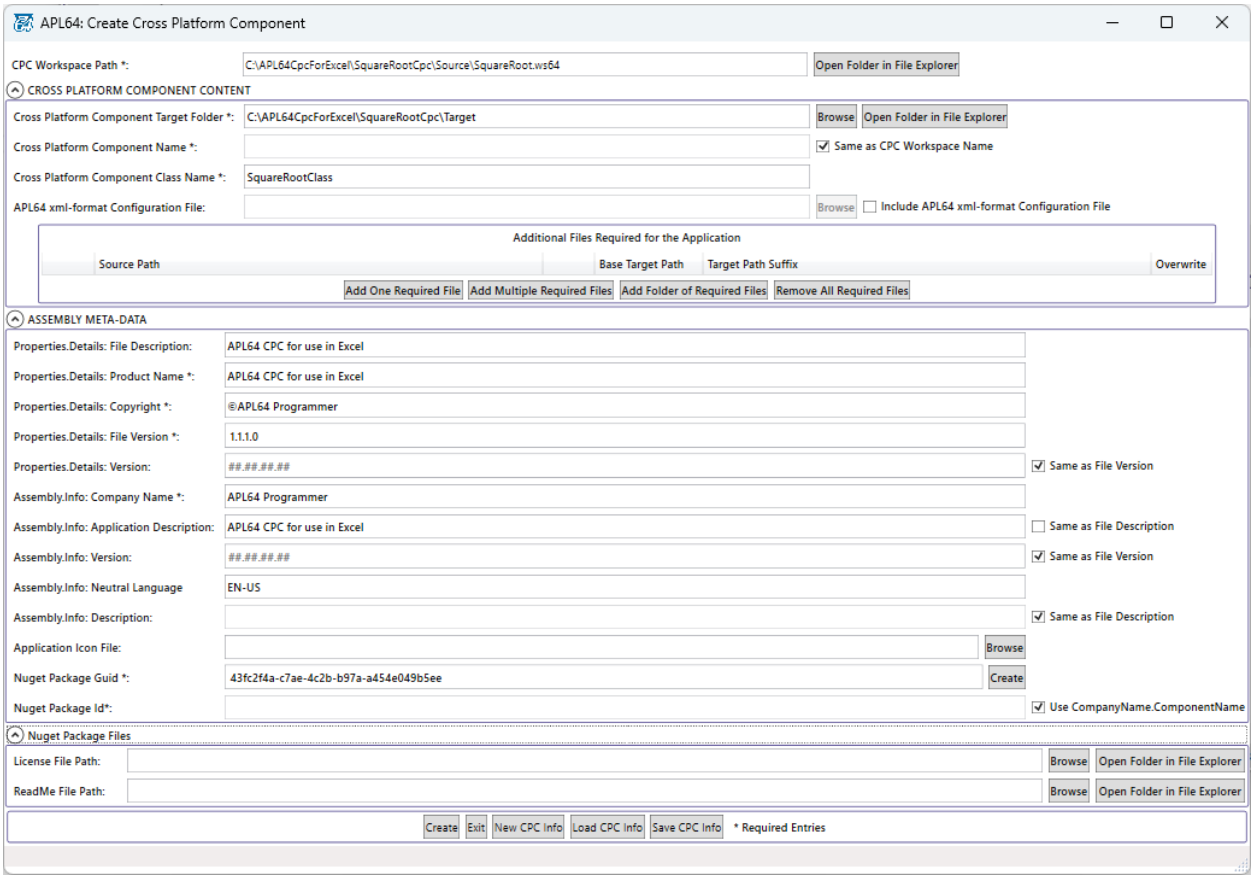

## Save the workspace to the Source Folder

The Source Folder is a sub-folder of the c:\ APL64CpcForExcel\SquareRootCpc\ folder for this project.

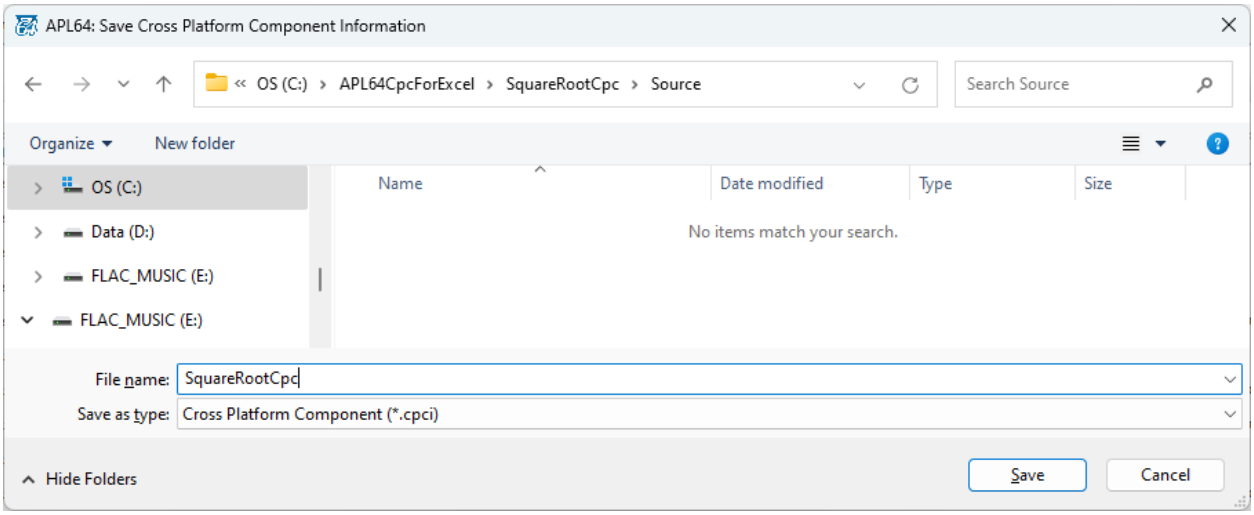


## Using the APL64 CPC creation utility, complete the APL64 CPC Creation Utility Dialog

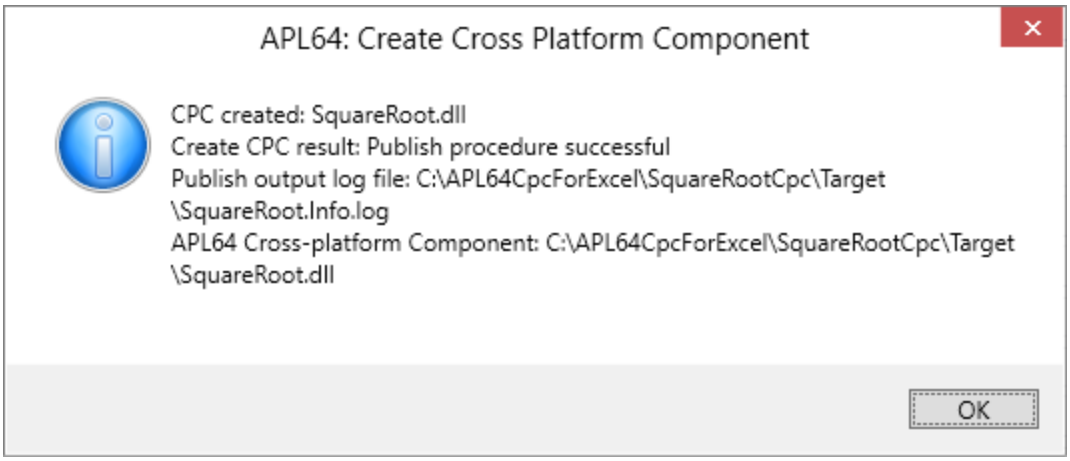





## Save the CPC Info to the Source Folder




Click the Create button to create the APL64 CPC Nuget package:



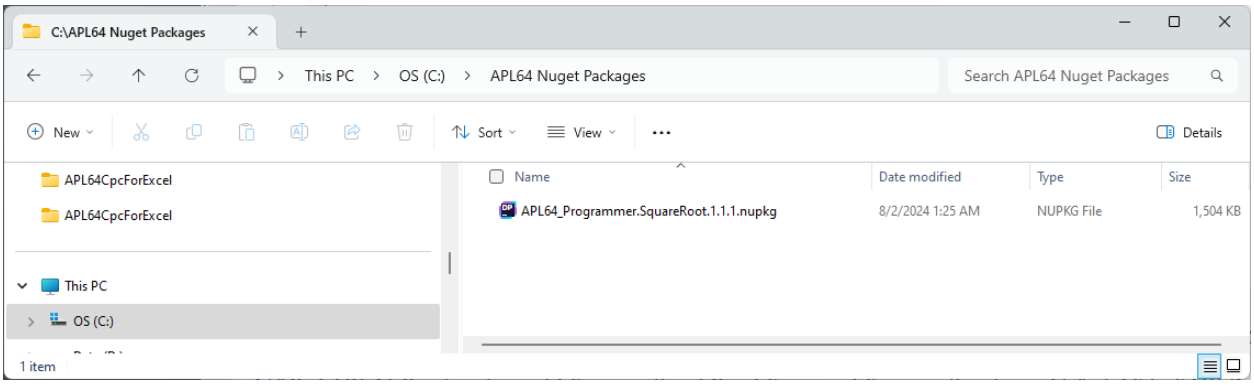

## Set up the Nuget Package Publication Folder

Set up Visual Studio to Recognize the Nuget Package Publish Folder as a Package Source

### Create the 'C:\APL64 Nuget Packages' folder on the workstation

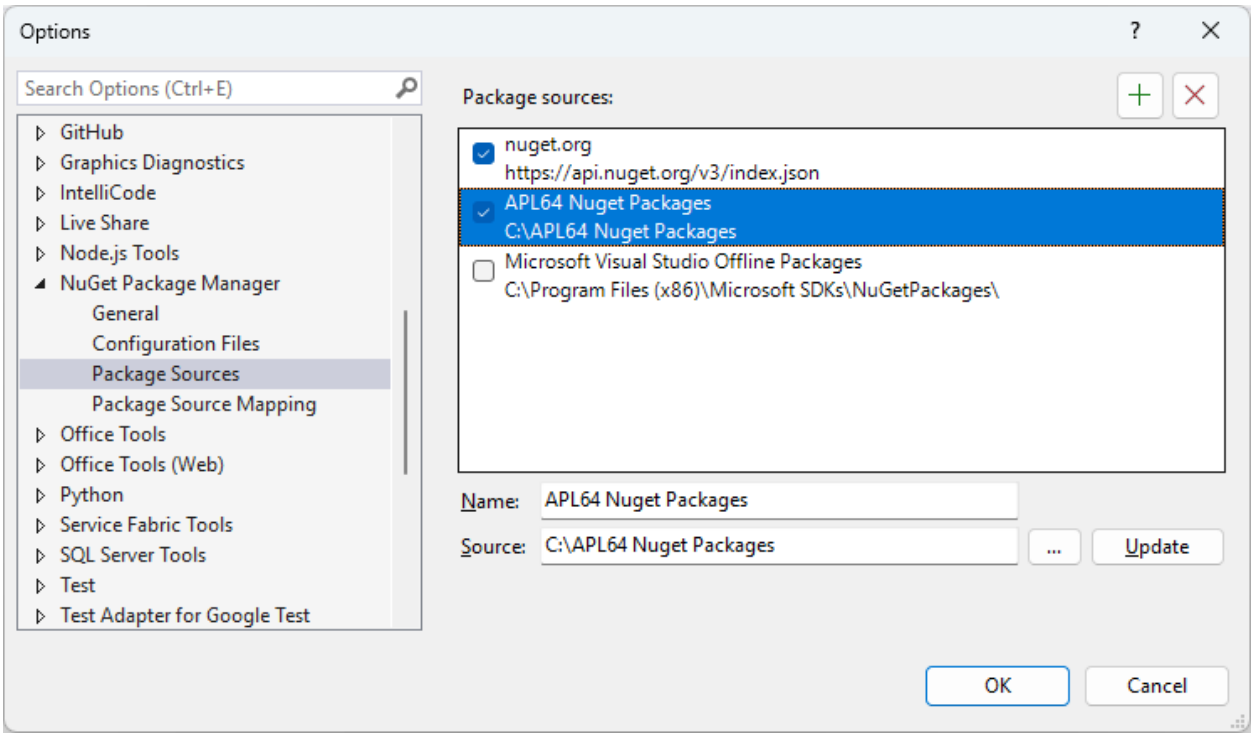
Nuget packages may be published on a web server or locally on a workstation. For development purposes, a local workstation folder (C:\APL64 Nuget Packages) is used.

Do not use the sample application folder (C:\APL64CpcForExcel\SquareRootCpc\Target) for this purpose, as this may include spurious components into the CPC.




### In Visual Studio identify the C:\APL64 Nuget Packages\ folder as a Nuget Package Source

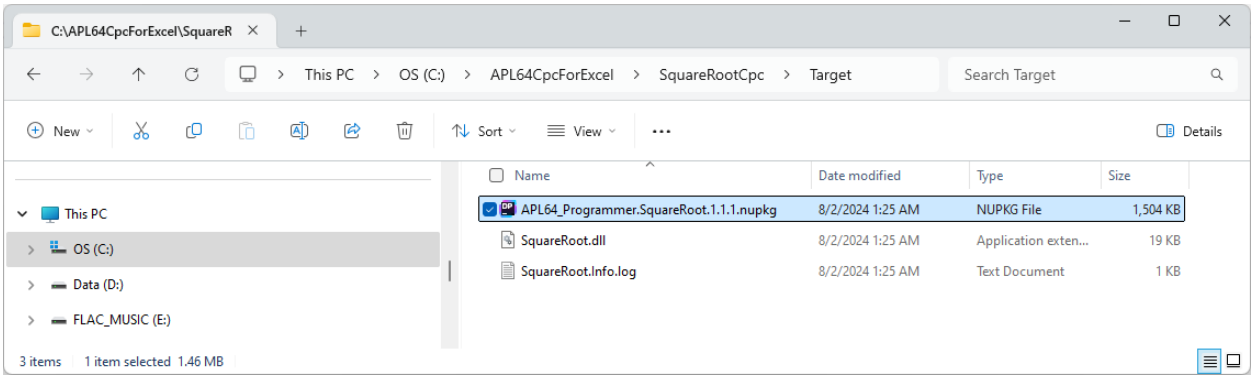
Use the Visual Studio Tools | Nuget Package Manager dialog to recognize the folder as a Package Source for Visual Studio:



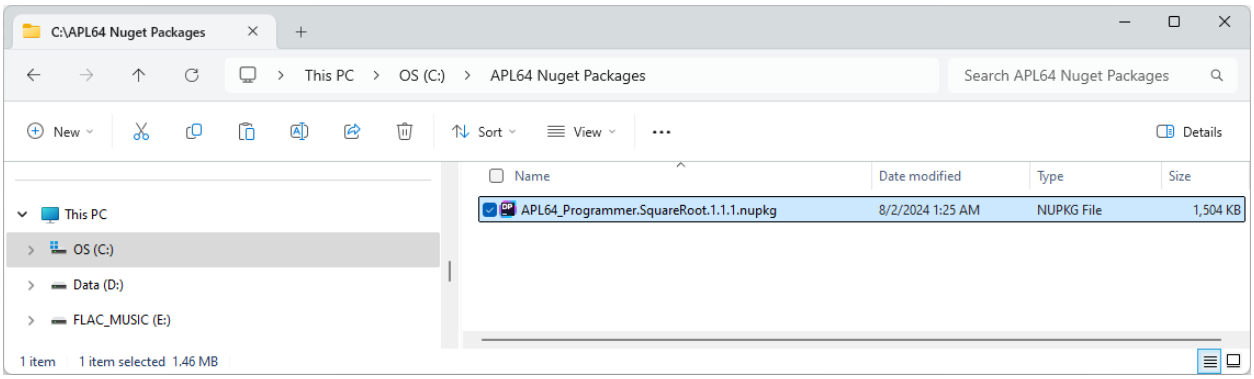

## Copy the APL64 CPC to the Nuget Package Publication Folder

Copy the APL64 CPC Nuget package for SquareRoot from the CPC Target folder to the Nuget package publication folder:

From:



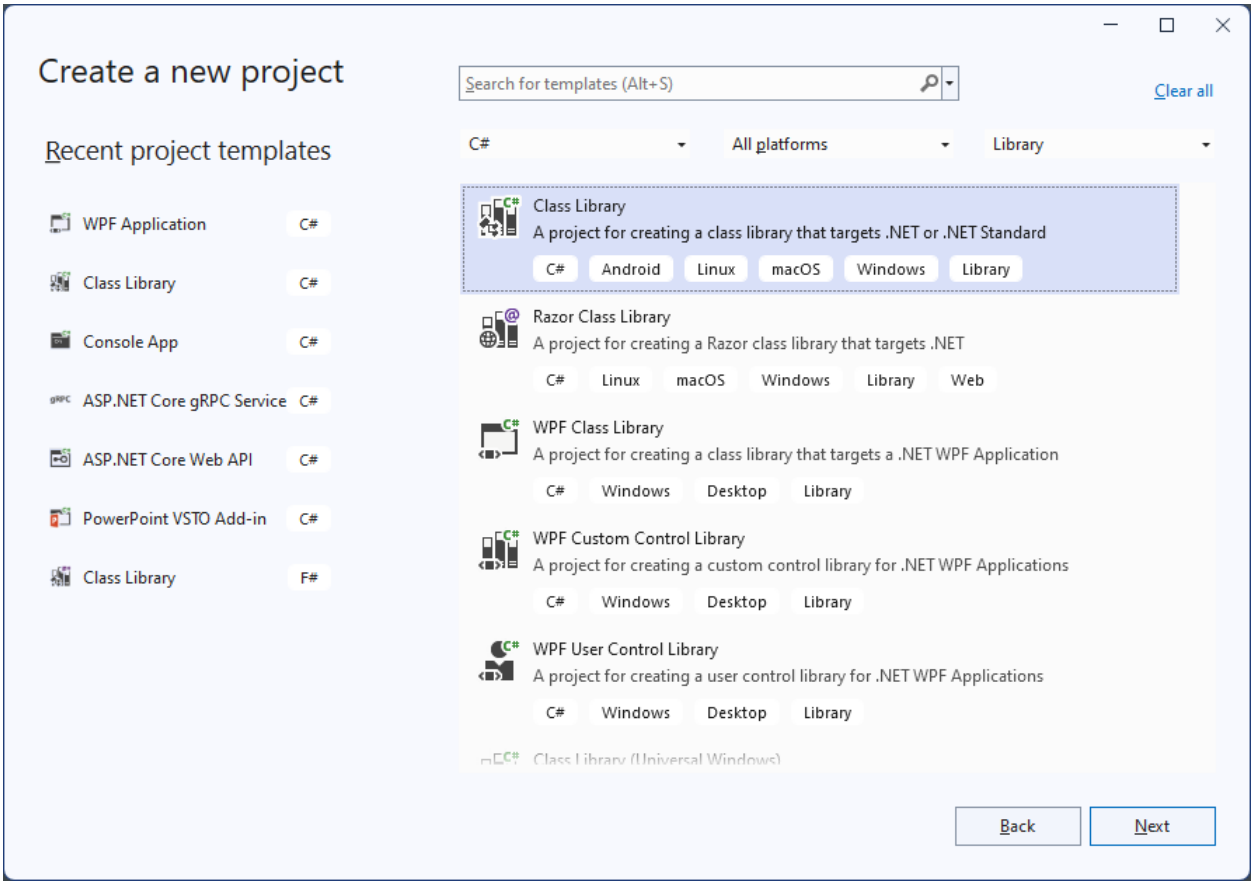

To:




# Create a .Net Application in Visual Studio

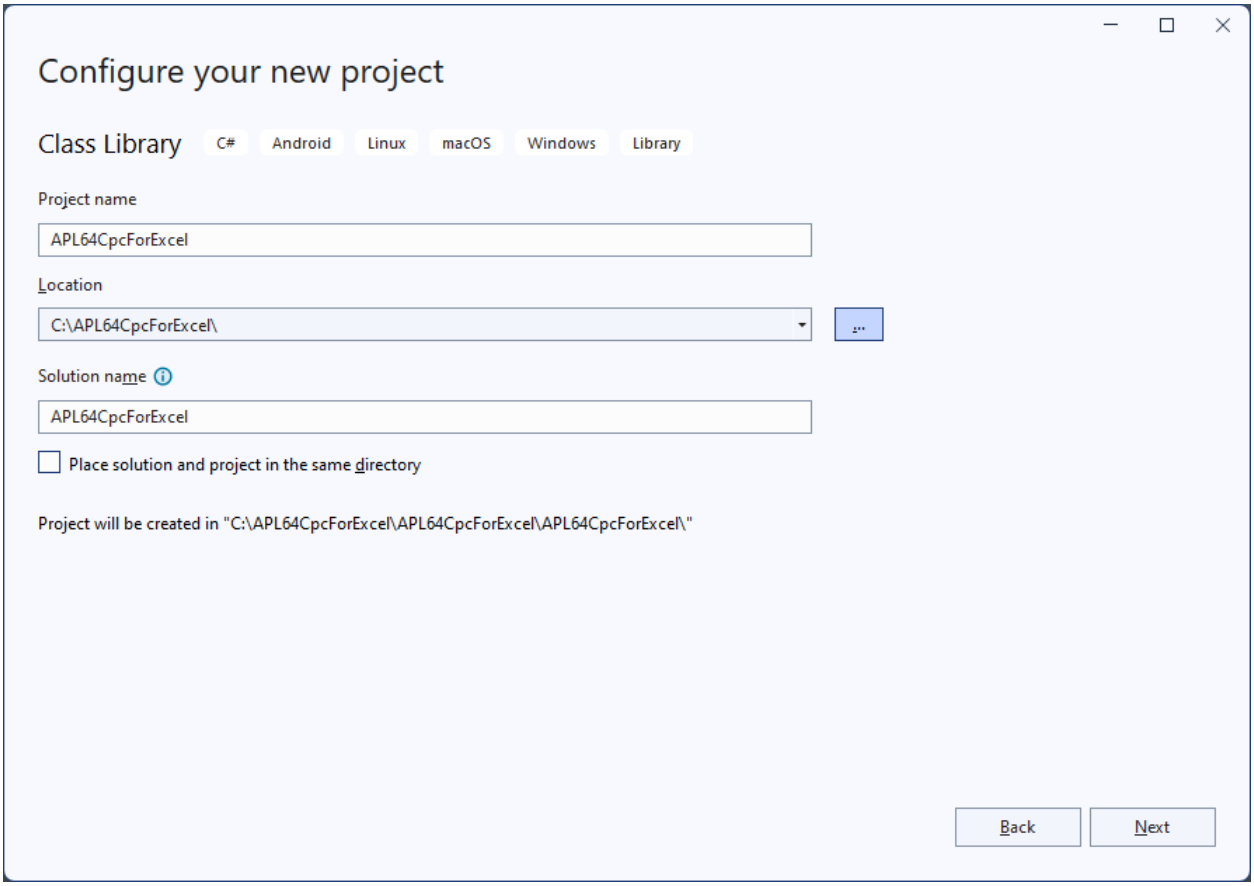
A .Net class library project is created which will contain the APL64 CPC so that the public functions in the APL64 workspace are available to Excel.

## Select the 'library' project type for .Net or .Net Standard




## Configure the project

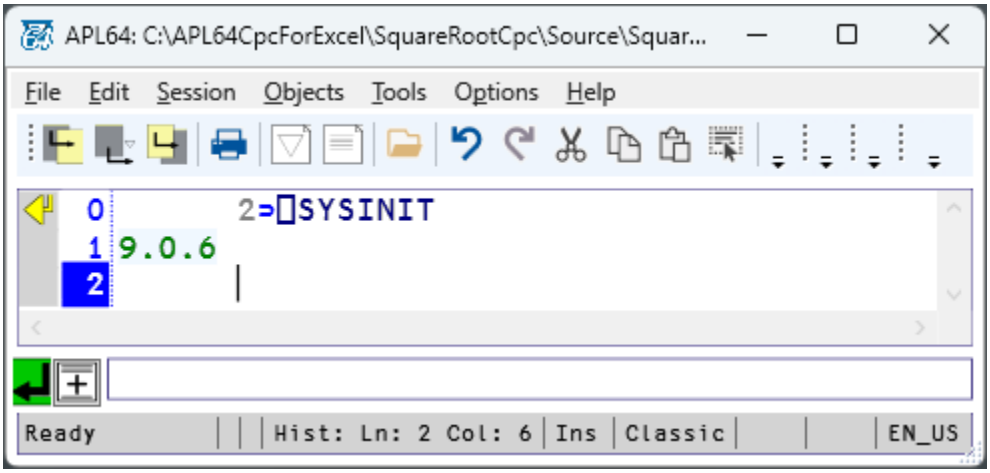
Specifying the name of the solution and the folder to contain the .Net solution:

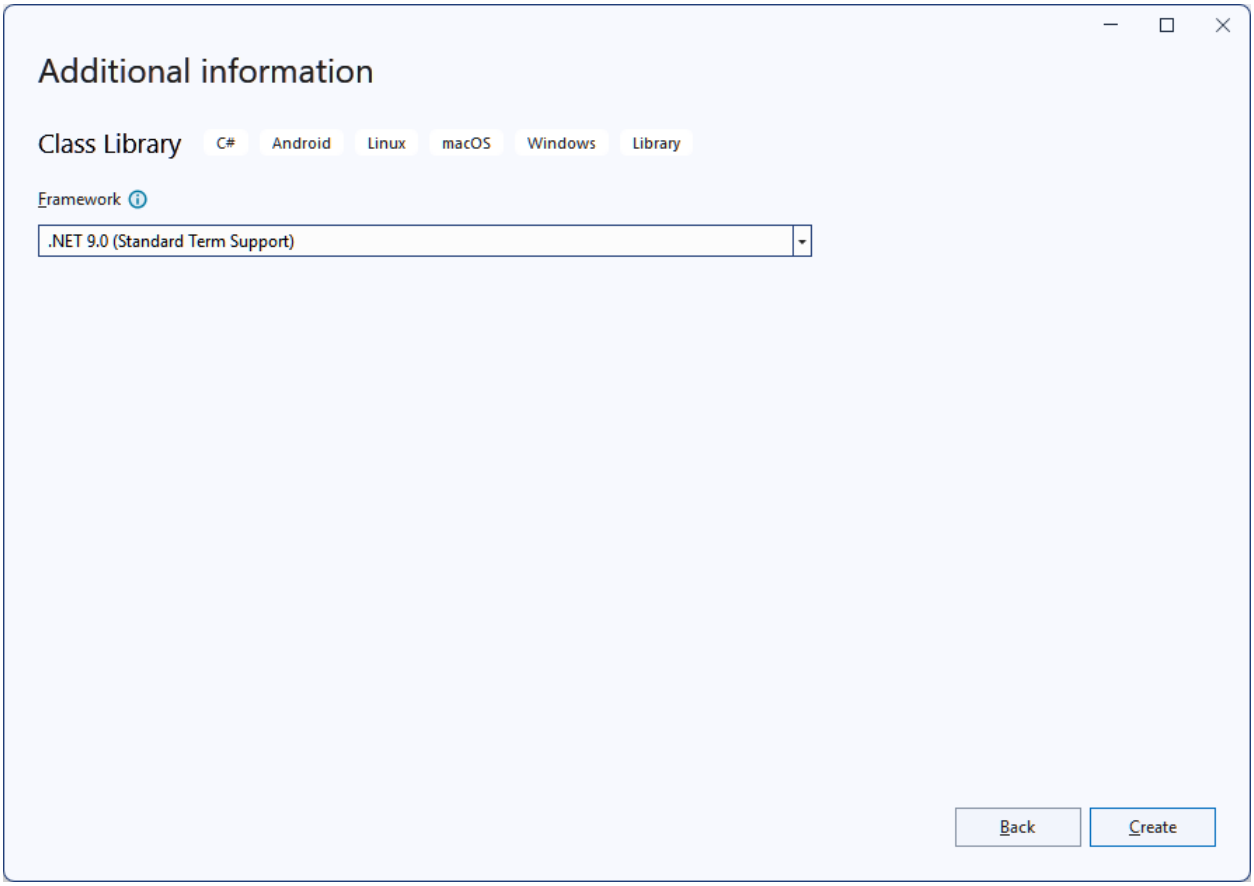



## Provide additional information for the project

### Select the .Net version

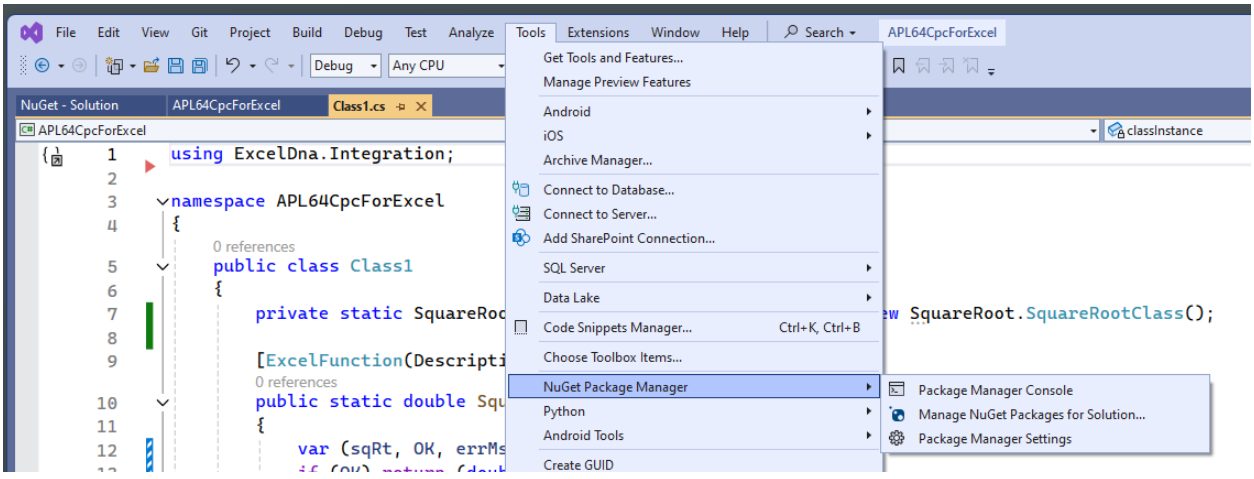
The .Net version for APL64 can be obtained using 2⊃□SYSINIT:






## Install the Nuget packages for the solution

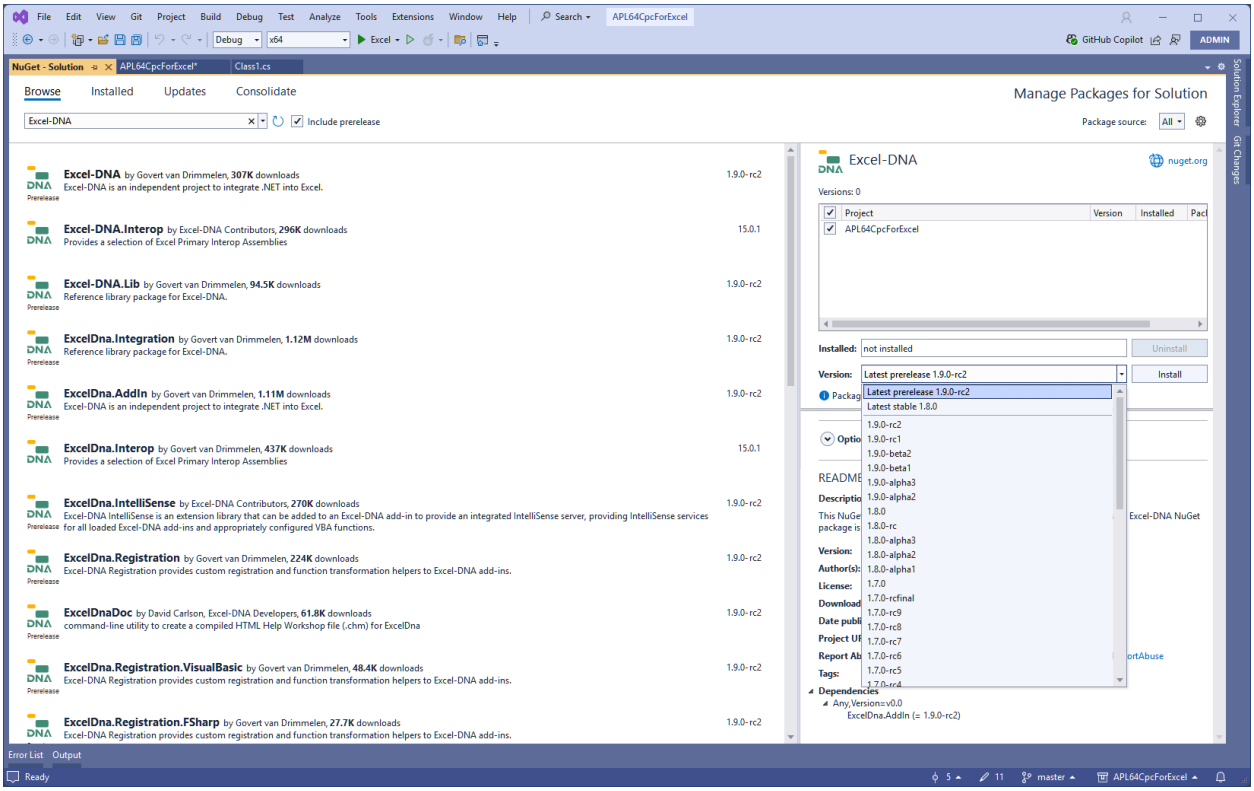
Use the Visual Studio Tools | Nuget Package Manager | Manager NuGet Packages for Solution dialog.



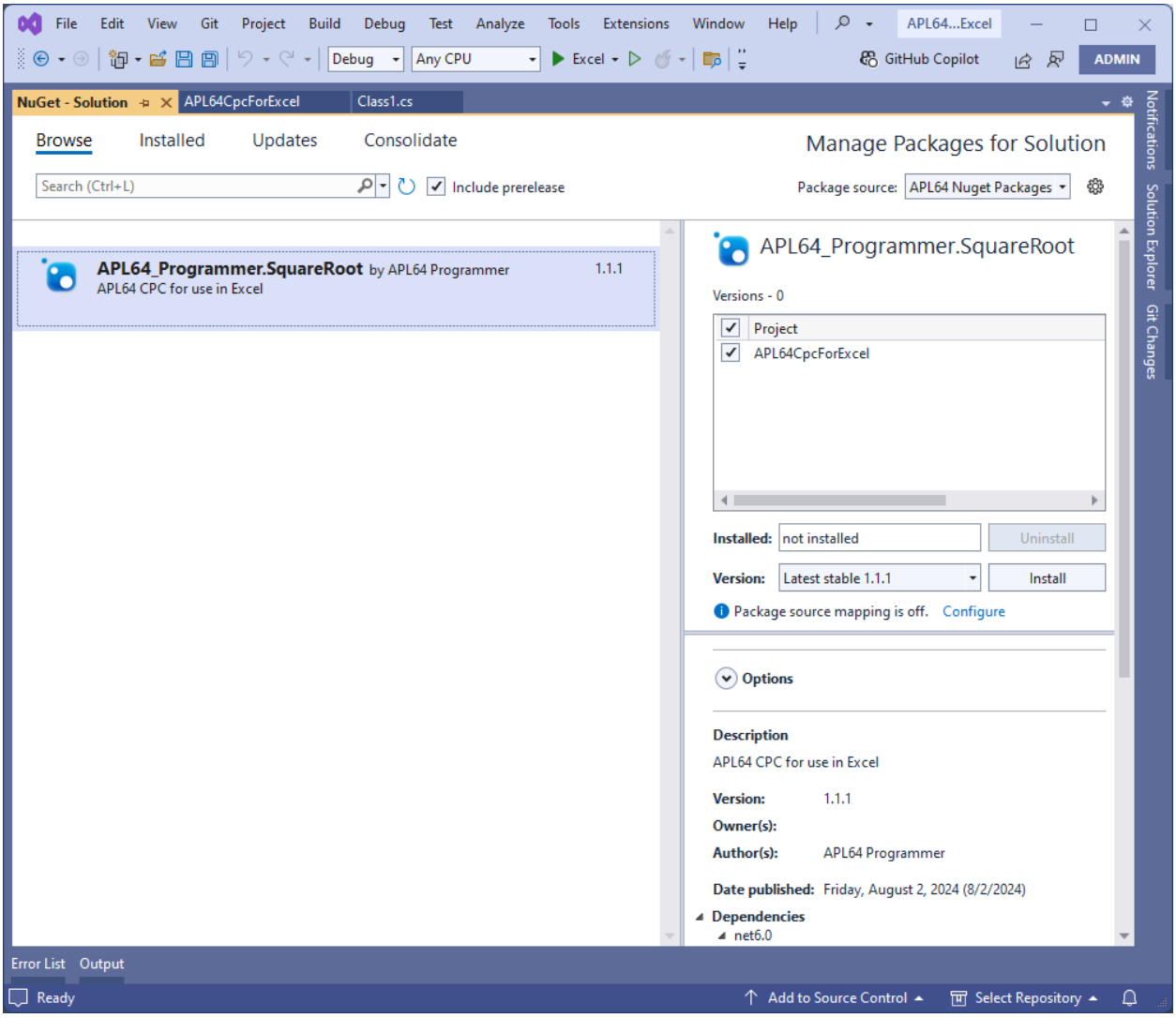

## Install the Excel-DNA Nuget package into the solution

The open-source, no-cost, .Net Excel-DNA tool is used to make the 'connection' between Excel and an APL64 CPC Nuget package. At the time this document was prepared, the developer of Excel-DNA has not released a production version which is compatible with .Net 9. For production use of the Excel-DNA Nuget package, it would be best to wait for a new production v1.9+.

For purposes of testing, the Excel-DNA v1.9.0-rc2 prerelease Nuget package can be used. In the Visual Studio Nuget package Browse dialog, check the 'Include Prerelease' check box, and install the Excel-DNA v1.9.0-rc2 version.



## Install the APL64 CPC Nuget package for SquareRoot into the solution




## Provide the C# Source Code for Class1 class in the Solution

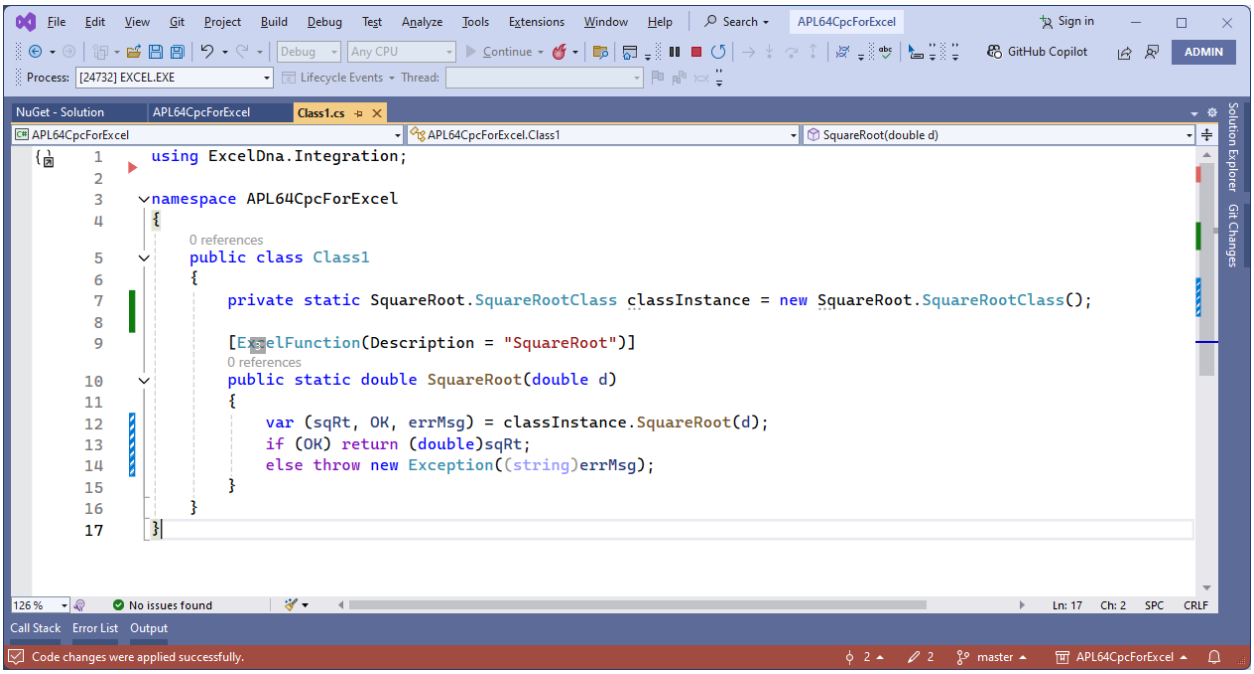
Replace the Class1 source code in Visual Studio with this code:

```
using ExcelDna.Integration;

namespace APL64CpcForExcel
{
   public class Class1
   {
      private static SquareRoot.SquareRootClass classInstance = new SquareRoot.SquareRootClass();

      [ExcelFunction(Description = "SquareRoot")]
      public static double SquareRoot(double d)
      {
         var (sqRt, OK, errMsg) = classInstance.SquareRoot(d);
         if (OK) return (double)sqRt;
```

```
        else throw new Exception((string)errMsg);
      }
    }
}
```



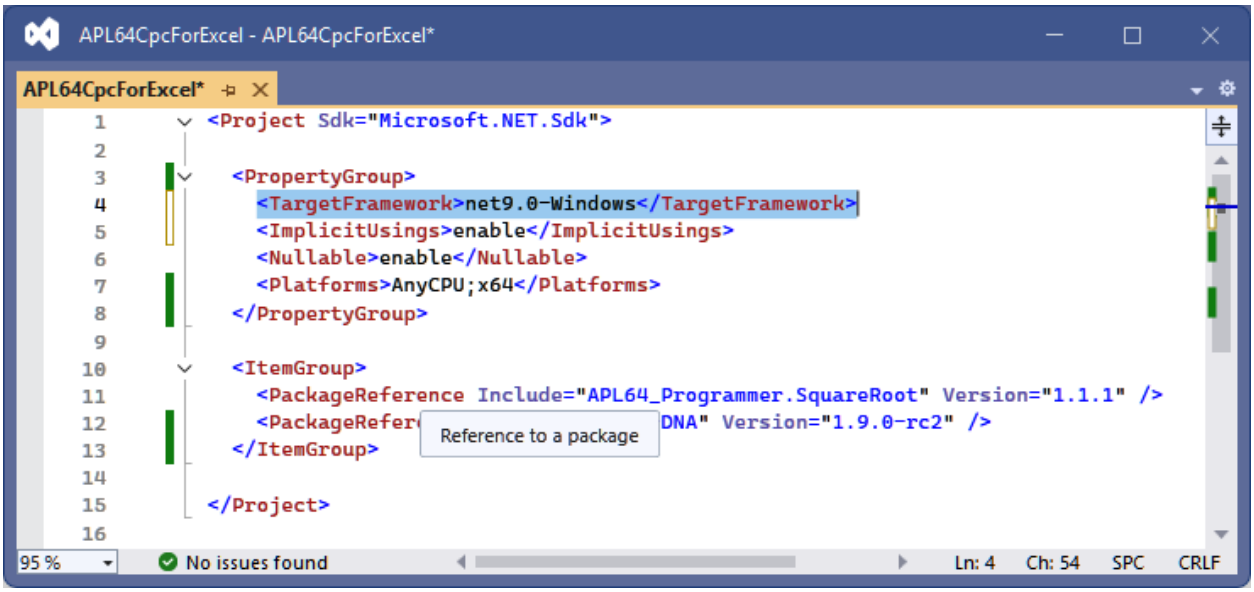

## Update the TargetFramework to the appropriate version

The appropriate version is available using 2⊃□SYSINIT in APL64. In the Visual Studio Solution Explorer double left click on the APL64CpcForExcel project to open thexml-format project file for editing and replace:

<TargetFramework>net9.0</TargetFramework>

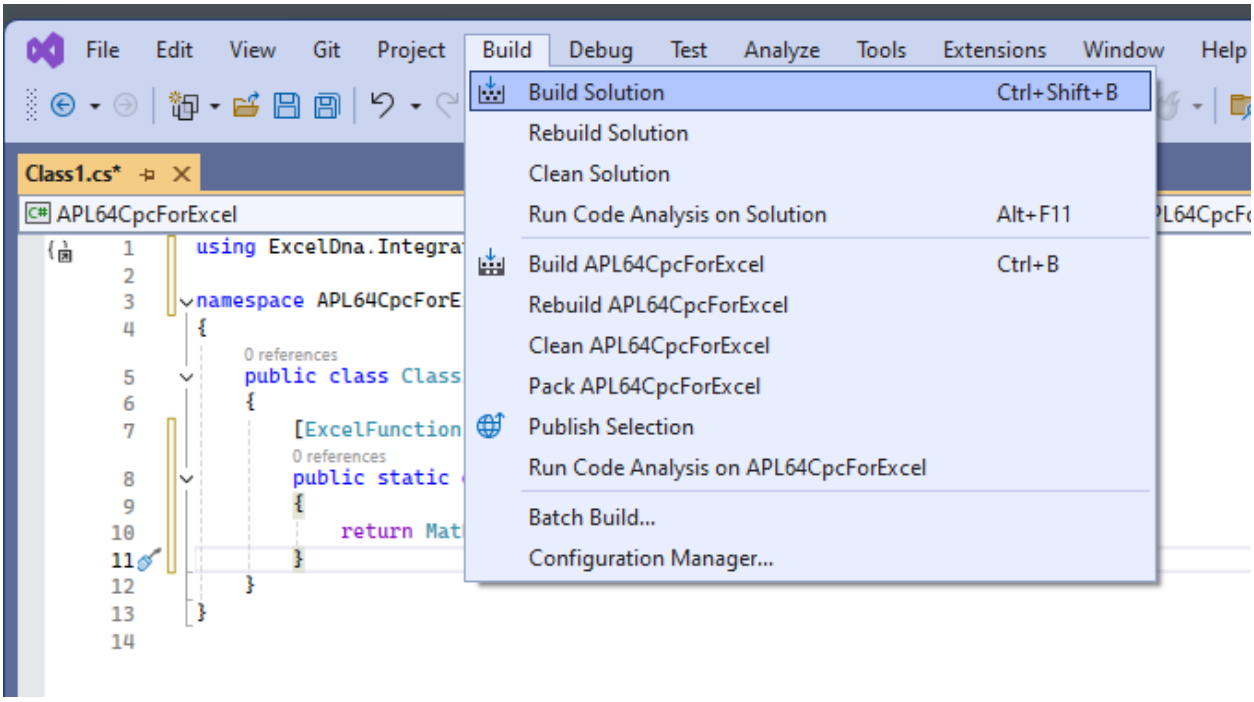
With:

```
<TargetFramework>net9.0-windows</TargetFramework>
```



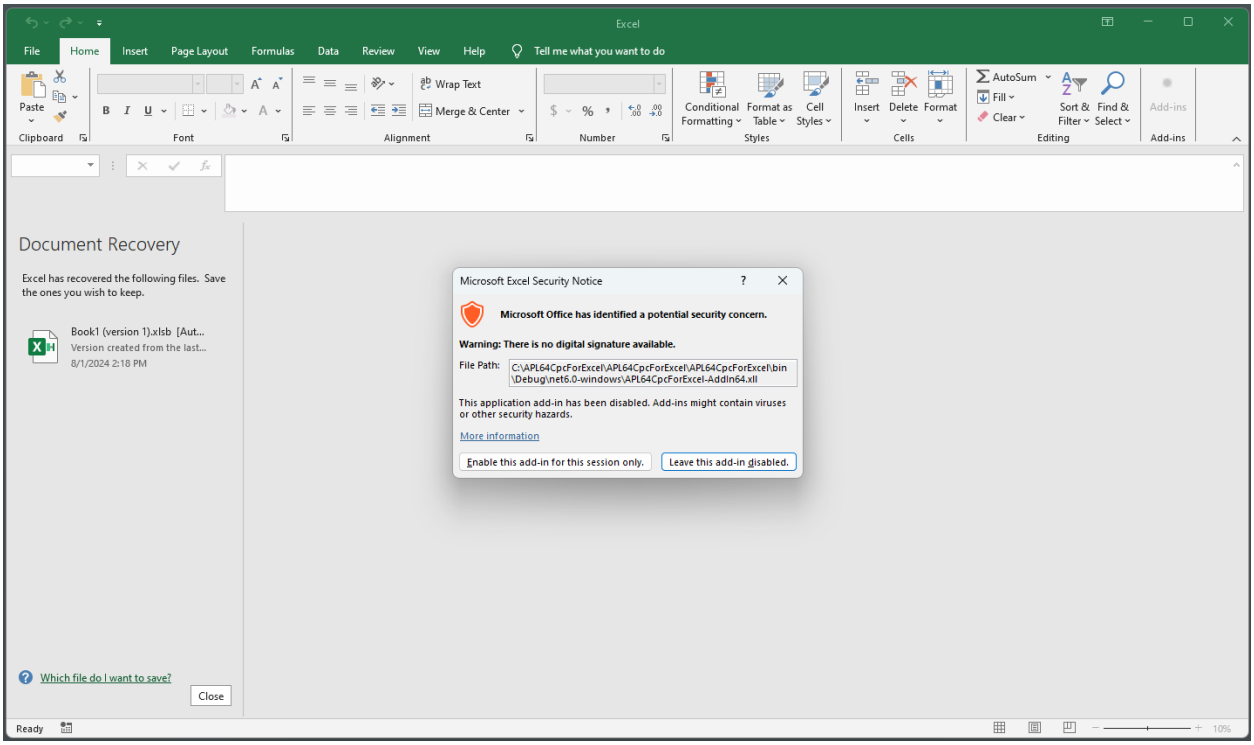

## Build the Visual Studio Solution

Use the Build | Build Solution menu item to build the solution.



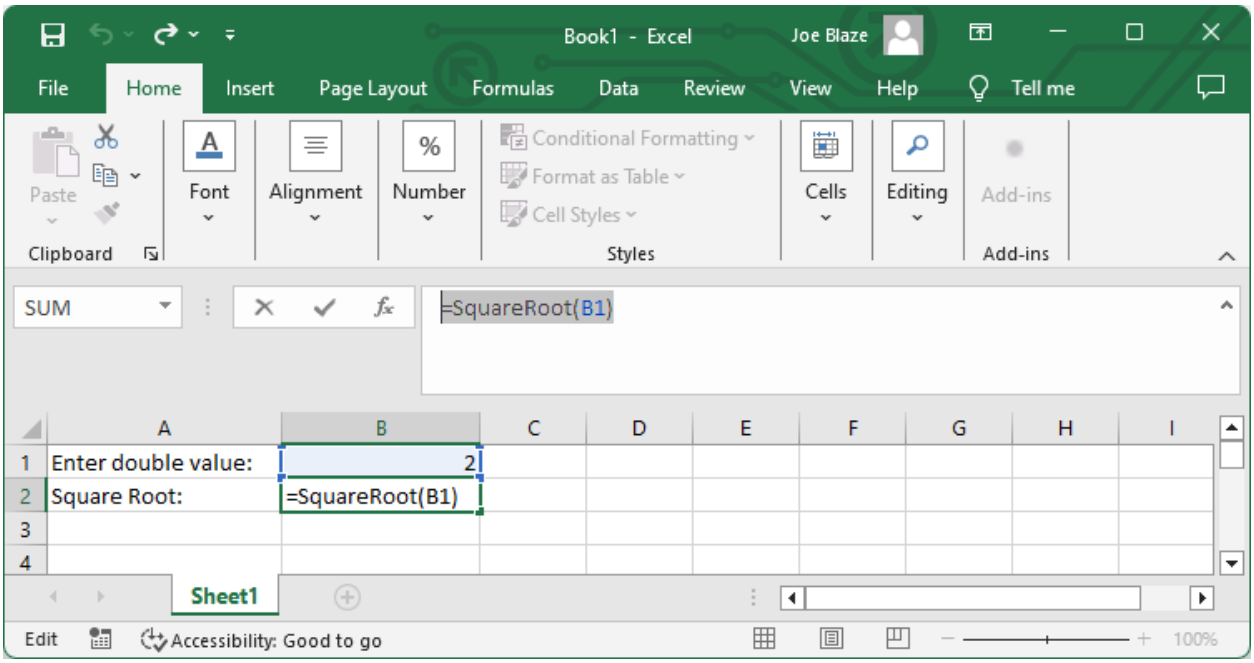

# Run the .Net Application

Run the .Net application (Click F5 or use the Debug | Start Debugging menu item). The application will start Excel and a security reminder will be presented. To continue with the application, click the 'Enable…' button:

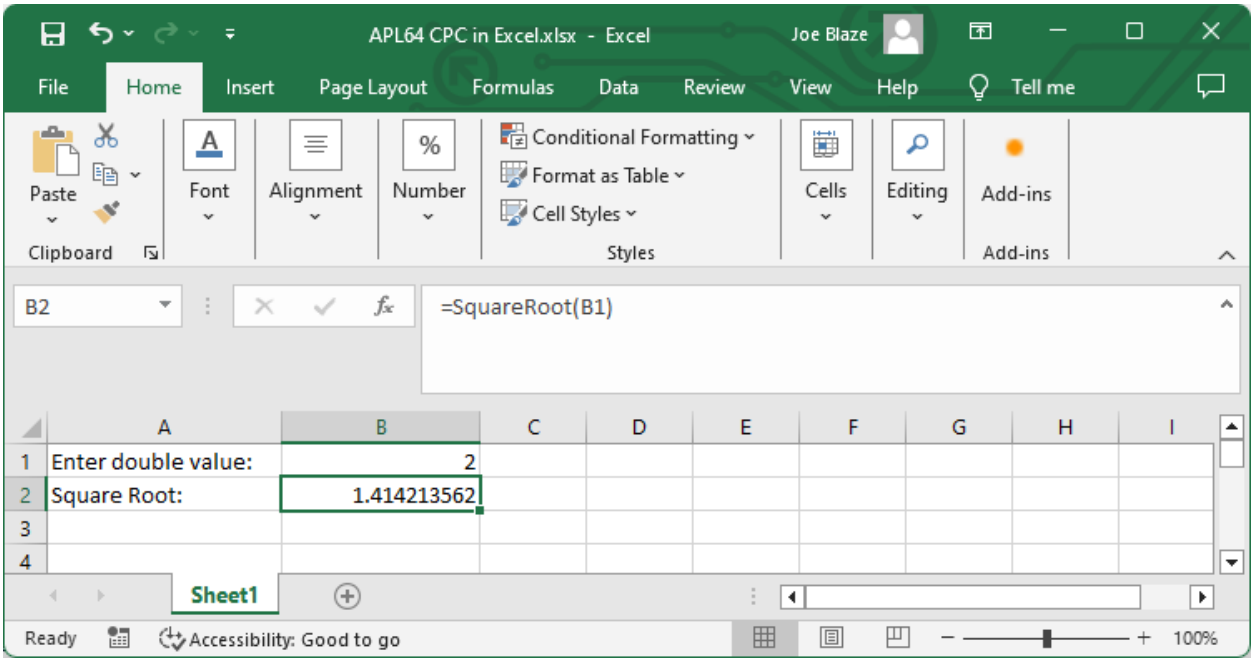



In Excel, create a new Excel workbook and use the SquareRoot function in Excel:

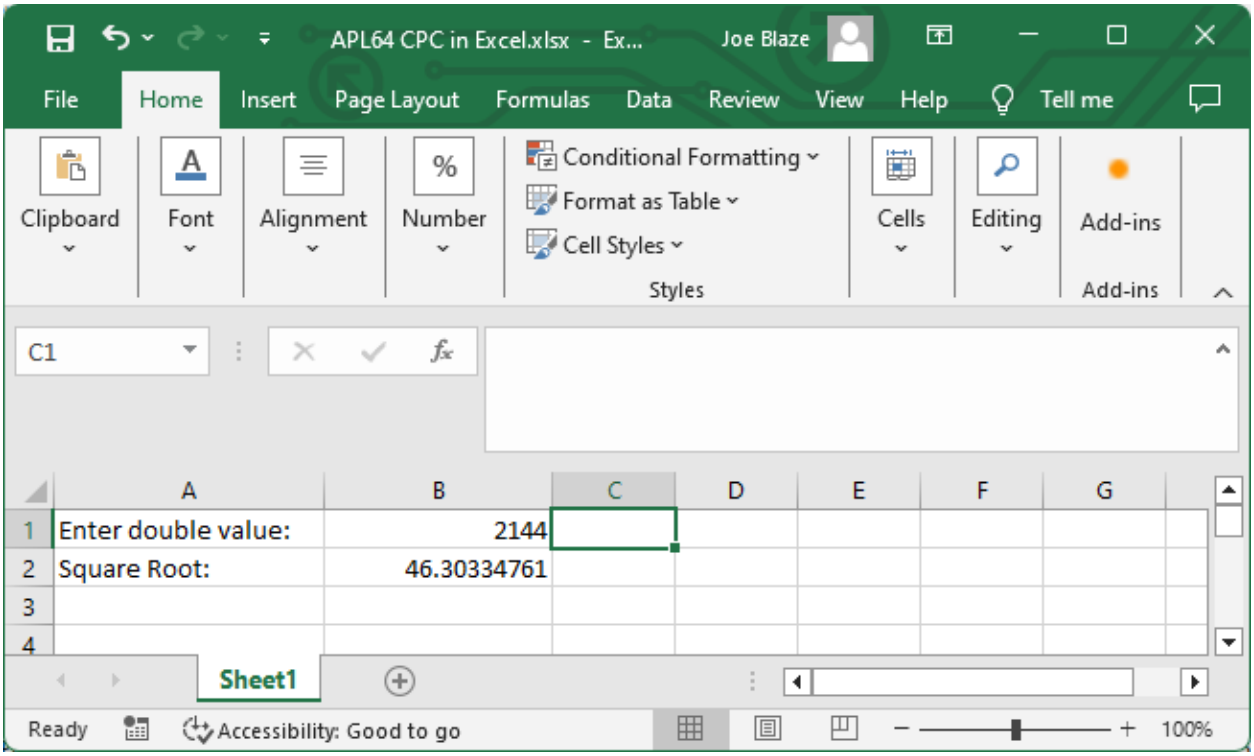
- Insert a prompt text is cell A1
- Insert a default value in cell B1
- Insert a description text in cell A2
- Insert the formula '=SquareRoot(B1)' in cell B2 so that the APL64 CPC SquareRoot function will be used to compute the square root of the value in cell A1



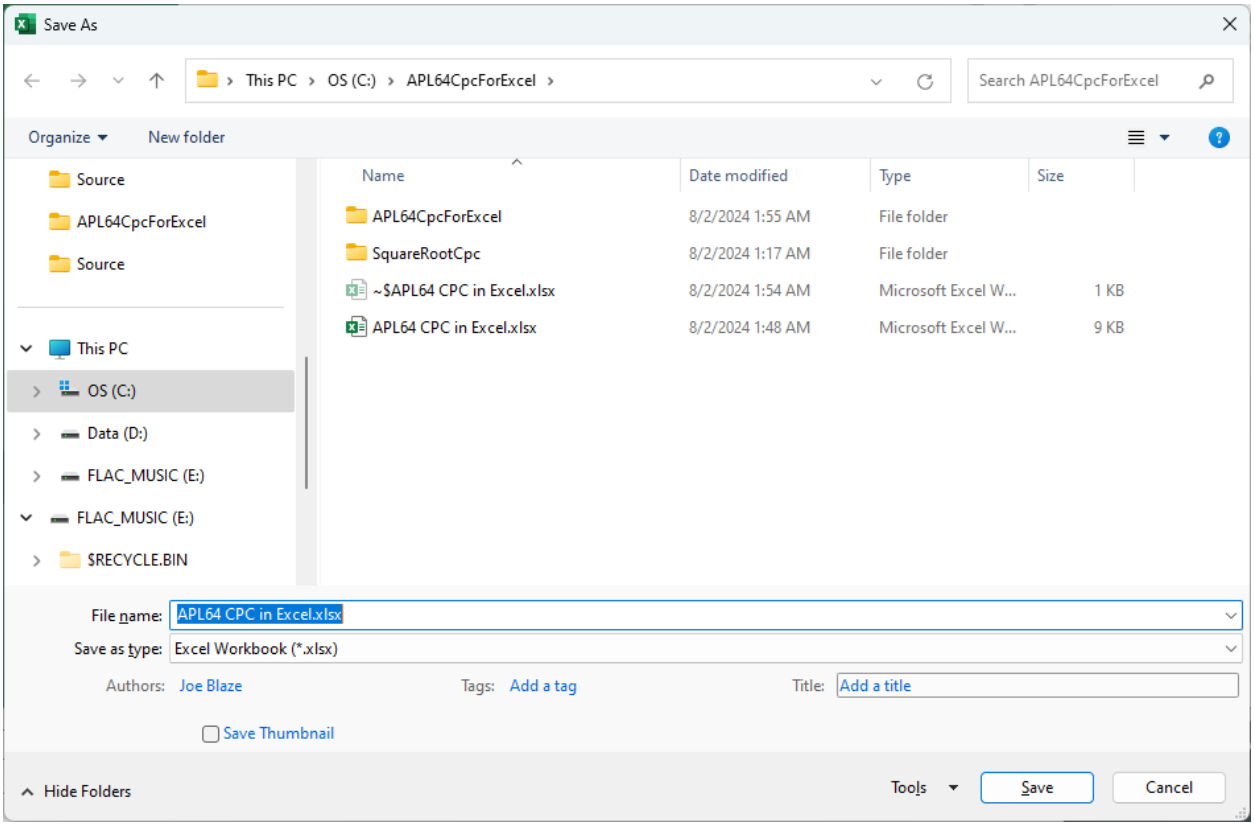
Move the keyboard from away from cell B2 and observe the APL64-computer result in cell B2:




Change the B1 entry and move the keyboard focus away from cell A2 to calculate a different square root:




Save the Excel workbook. When the .Net application is run and Excel starts, this workbook may be opened and the APL64 CPC SquareRoot function will be available in Excel.




## Deploying the Application

The example provided here is greatly simplified to illustrate the basic use of an APL64 CPC in Excel. A production-ready application involves more design and development in .Net to provide a professional result.

Some of the necessary deployment elements may include:

- User configuration options
- User GUI
- Application-specific Excel workbook(s)
- End-user documentation
- Link to the appropriate .Net runtime installer or the installer itself
- Digital signature for installer and application
- Other application-specific components

An APL64 CPC is cross-platform compatible, but the Excel-DNA toolkit targets the Windows desktop environment. Be sure to review the Excel-DNA Nuget package documentation for a full understanding of the many options and features of the toolkit.

The APL64 Development Team is ready to assist APL64 and .Net programmers to create professional applications. Contact sales@apl2000.com to schedule customized consulting or support@apl2000.com for technical assistance which is provided to APL64 subscribers.