APL64: CPC in a JavaScript Server

Contents

Overview

An APL64 CPC is a cross-platform Nuget package which exposes one or more APL64 programmer-developed public functions as .Net methods.

This example illustrates how .Net methods can be used from html/JavaScript web pages.

This document outlines the creation of an APL64 CPC and a JavaScript web page which uses the APL64 CPC. The reader of this document is assumed to have a significant understanding of APL64 CPC creations, JavaScript, html web pages and web servers.

Download the Project

This document is not step-by-step recipe.

Download the entire project here. After downloading the zip-format file, expand it to c:\CpcJs to match the references in this document. To run the project, follow the instructions in this document which require creating the APL64 CPC and referencing it in the .Net Bridge Process project.

Prerequisites

Components which must be installed on the development workstation:

- APL64 Developer version
- .NET SDK (1)
- Node.js (2)

Components which must be installed on the server workstation:

- .NET SDK (1)
- Node.js (2)

Components which must be installed on the client workstation:

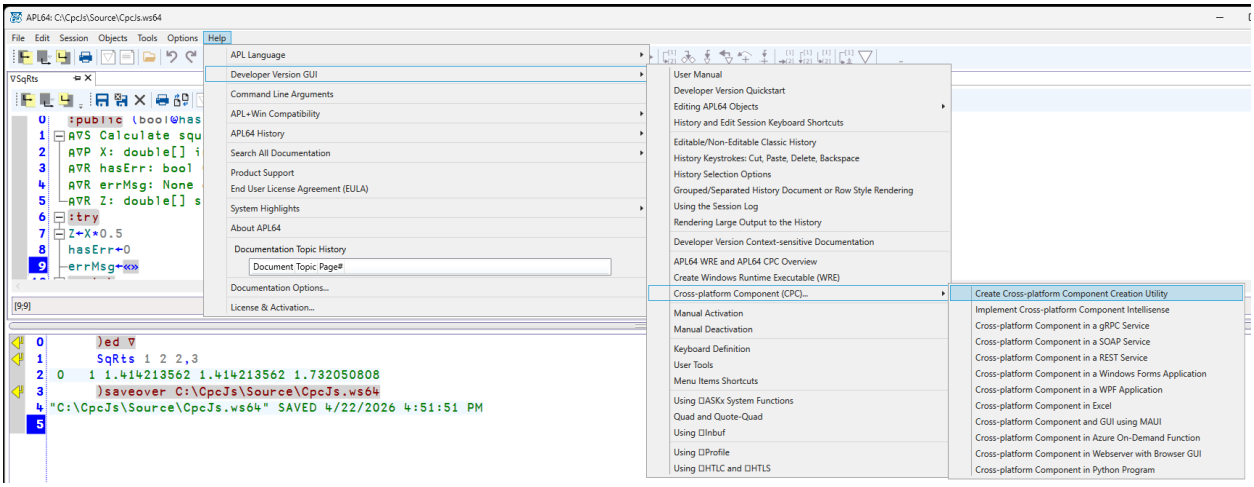
- Up-to-date Internet browser

(1) .Net SDK is a no-cost component available here. Use 2⊃□sysinit to determine the APL64-compatible .Net SDK version.
(2) Node.js, v24.14 or later, is a no-cost component available here.

Create the APL64 CPC

For details on the APL64 CPC creation utility and workflow, see the user documentation in the APL64 developer version menu item:

Help | Developer Version GUI | Cross-platform Component (CPC)... | ...Creation Utility

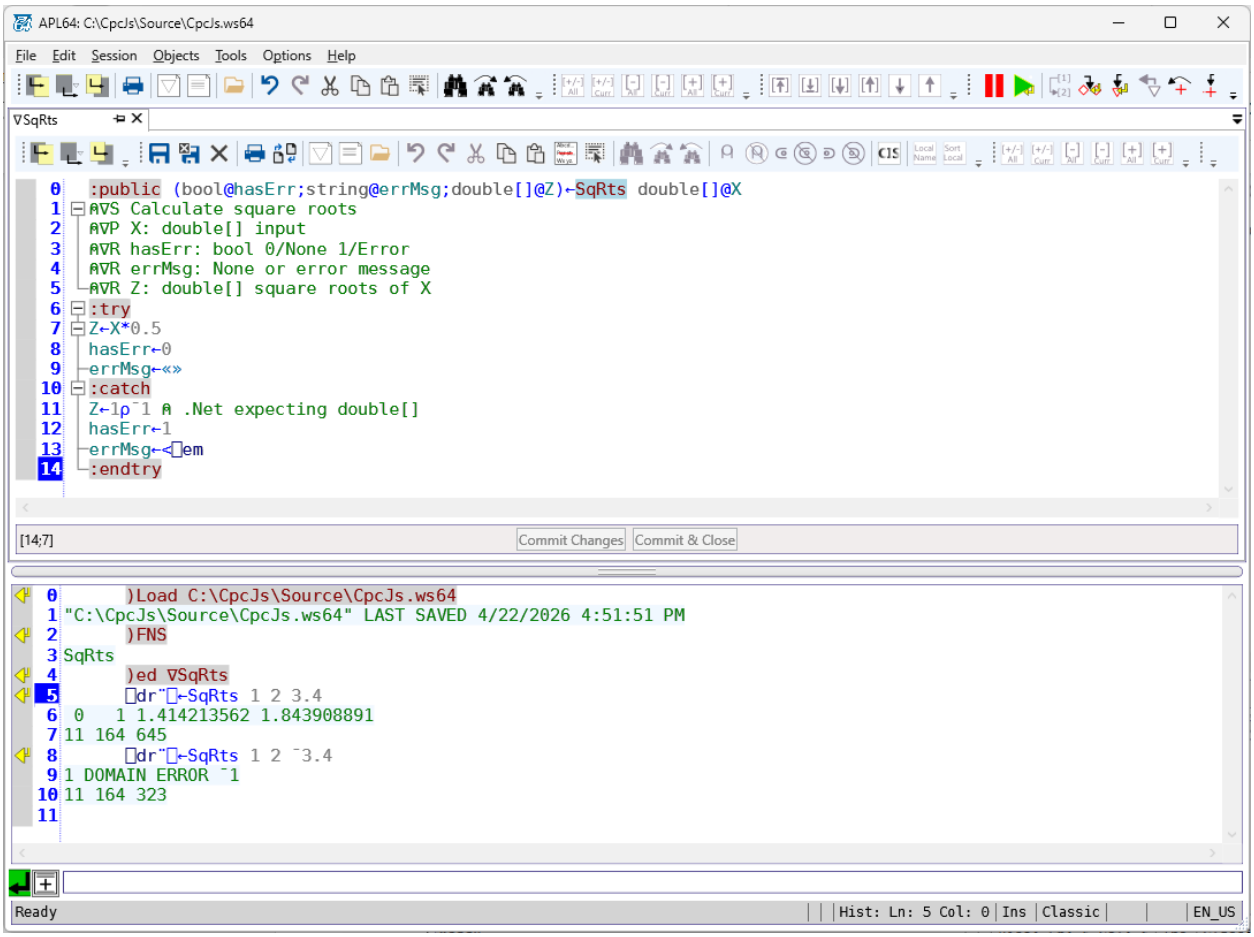


The APL64 public function

The SqRts public function is saved in the workspace: C:\CpcJs\Source\CpcJs.ws64.

The public function includes the Intellisense scaffold (public comments ⍝∇S, ⍝∇P, ⍝∇R), so that the APL64 CPC will provide Intellisense documentation for the end user of the APL64 CPC in the .Net environment.

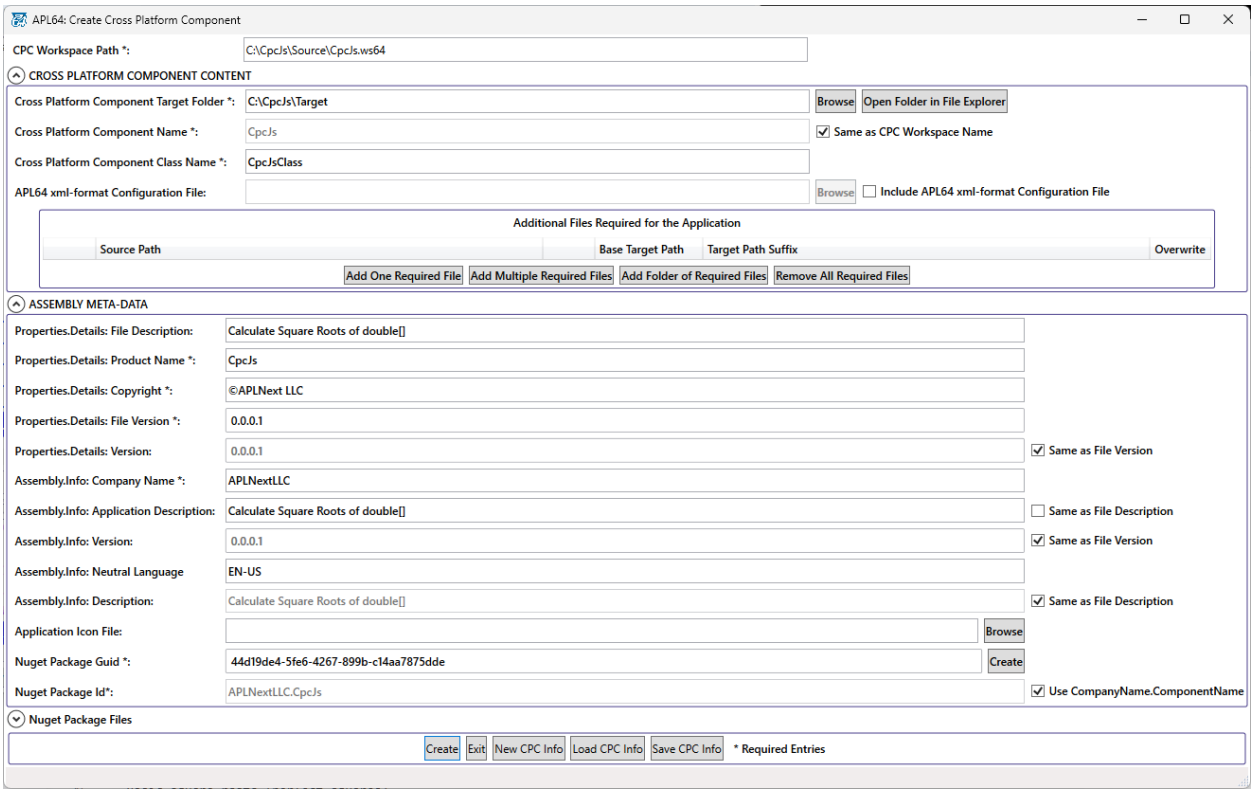
The public function can be tested in the APL64 developer version environment:



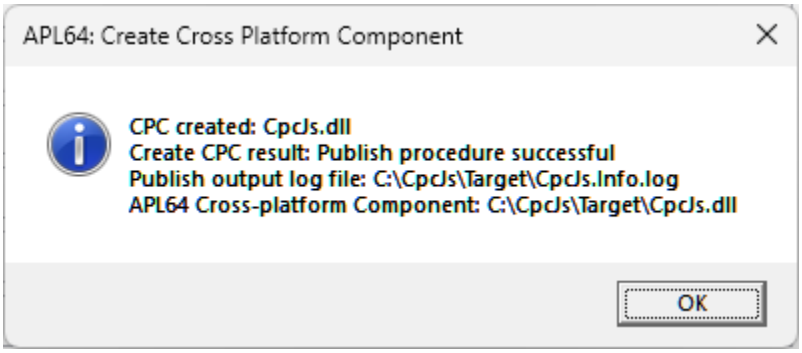
Complete the CPC creation utility dialog

The C:\CpcJs\Target folder will contain the APL64 CPC.

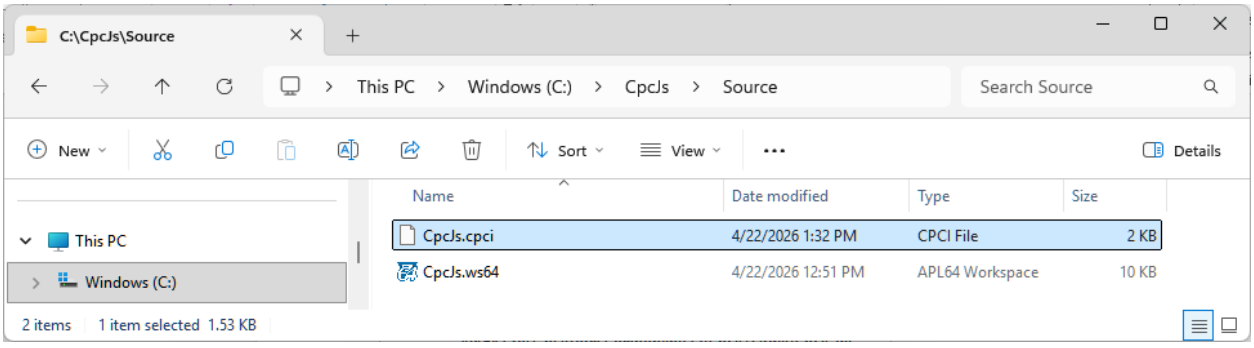
The completed CPC creation dialog:



Clicking the 'Create' button will create the APL64 CPC:

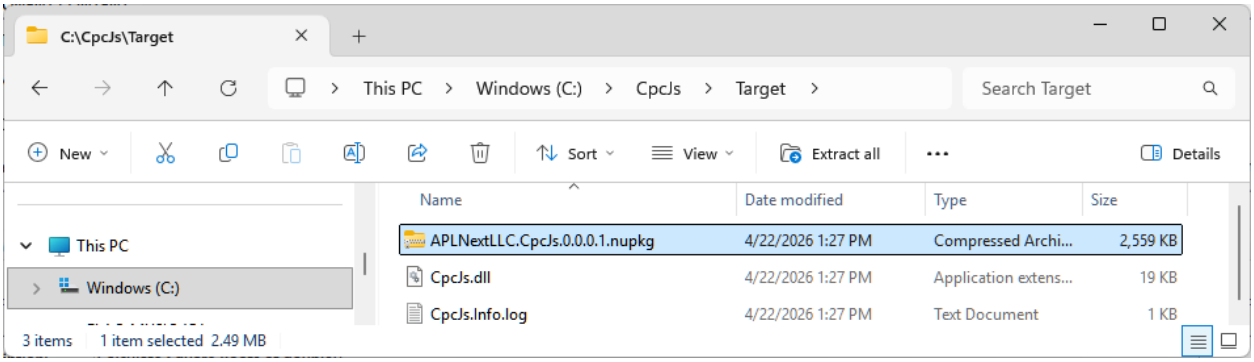


The information in the CPC creation dialog, CpcJs.cpci, is saved for future use in the C:\CpcJs\Source folder

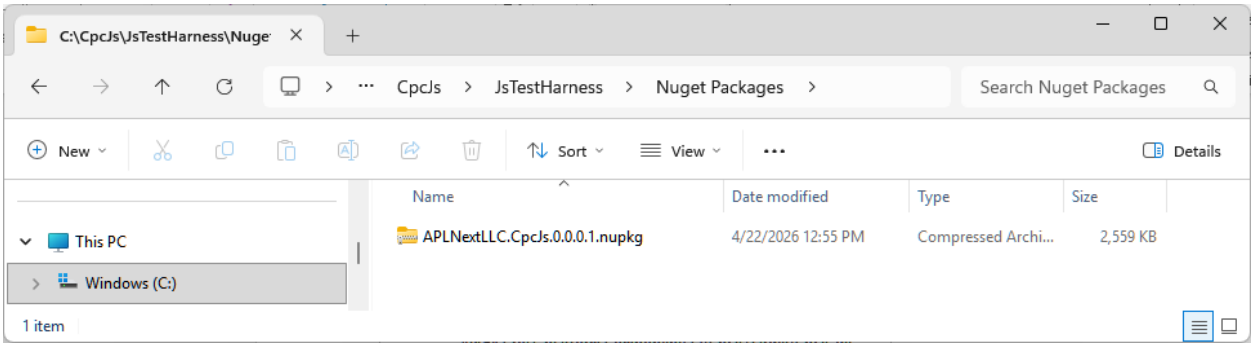


The C:\CpcJs\JsTestHarness\Nuget Packages folder contains the published APL64 CPC. In a production environment the CPC can be delivered to the end user or published on a Nuget package publish website.

To avoid including extraneous files in the APL64 CPC, the CPC in the target folder is copied from the target folder:



to the c:\CpcJs\JsTestHarness\Nuget Packages folder.



The JavaScript project

This document does not provide a step-by-step recipe to create the JavaScript application. Download the full project to see the project files referenced in this document.

Overall process:

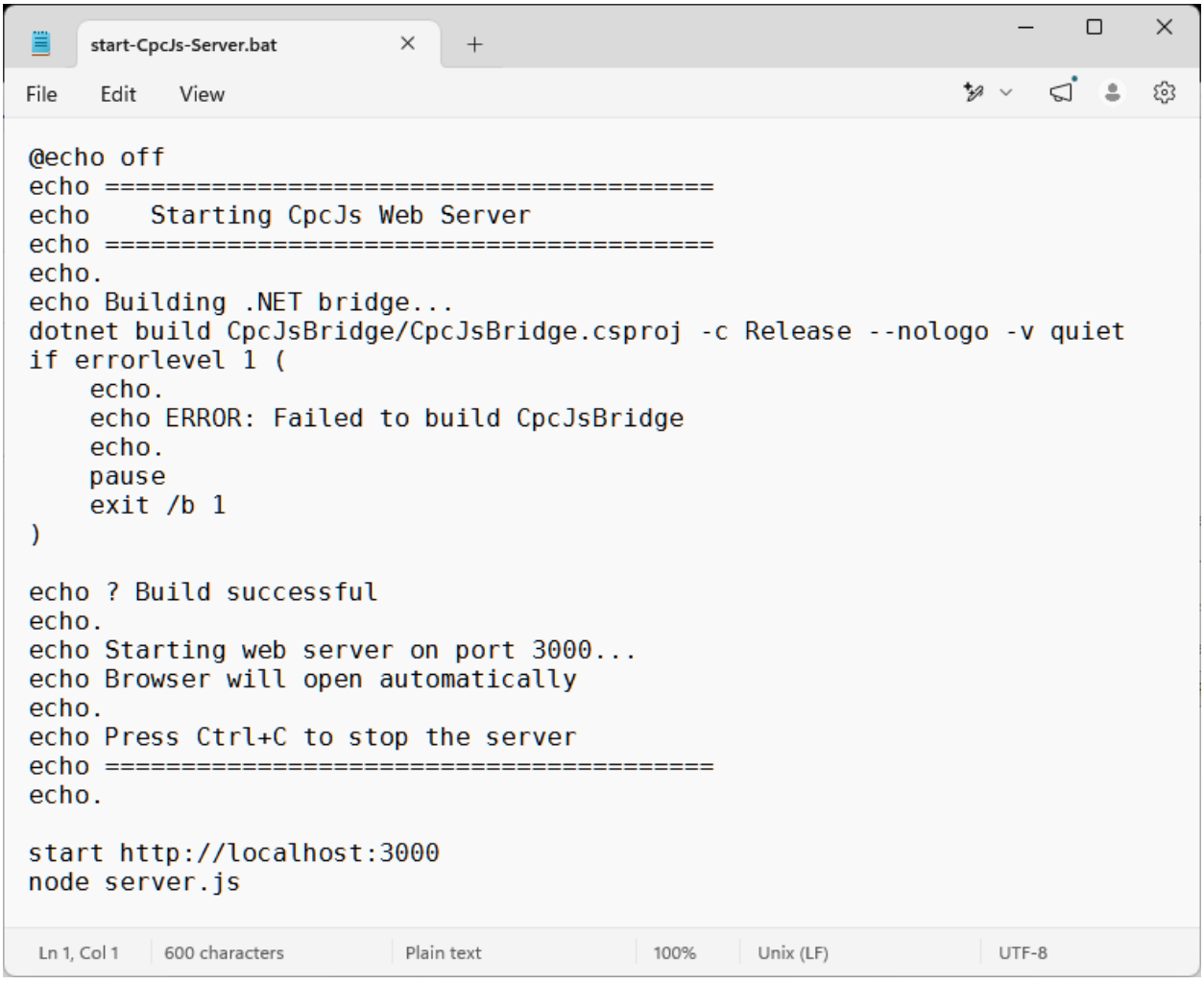
- The Start-CpcJs-Server.bat file is run to start the JavaScript server and open the default browser on the client workstation
- When the server.js starts, the index.html page is available to a client to access the server
- When the client inputs a double[] vector and requests server processing, the Node.js wrapper receives the input, asynchronously calls the .Net CpcJsBridge project, and waits for the result

- The CpcJsBridge project uses the APL64 CPC Nuget package SqRts() public function to calculate the square roots based on the client input.
- The results are passed to the client reversing the above flow of information.

Web server launcher

The C:\CpcJs\JsTestHarness\start-CpcJs-Server.bat file is run to start the server

- Builds the CpcJsBridge .Net project
- Starts the Node.js web server
- Opens the default browser to http://localhost:3000.

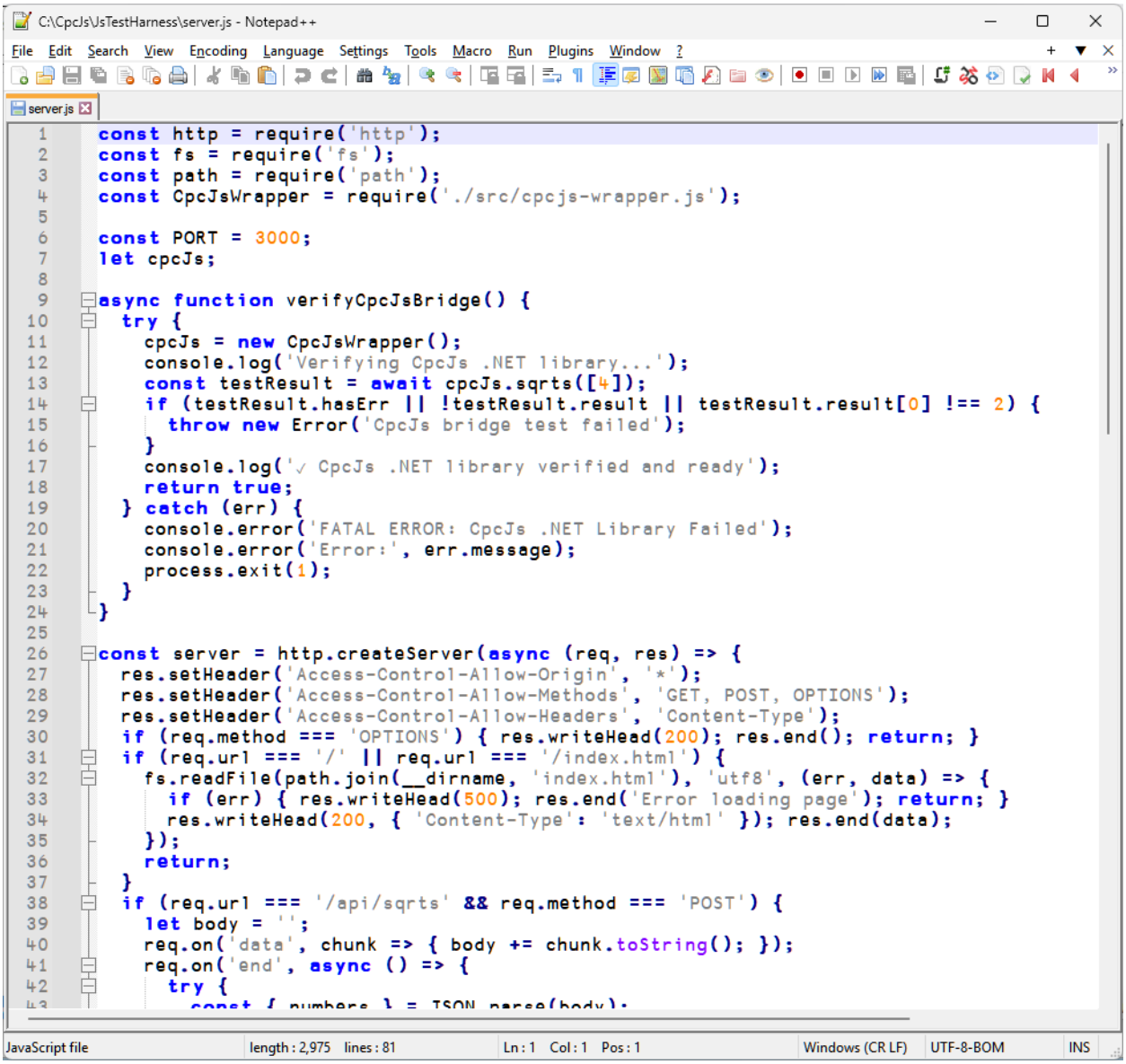


JavaScript web server

The JavaScript web page runs in a web server. For testing purposes, a JavaScript web server is created on the workstation using port 3000. In a production environment the application would be supported in a production web server.

The C:\CpcJs\JsTestHarness\server.js file:

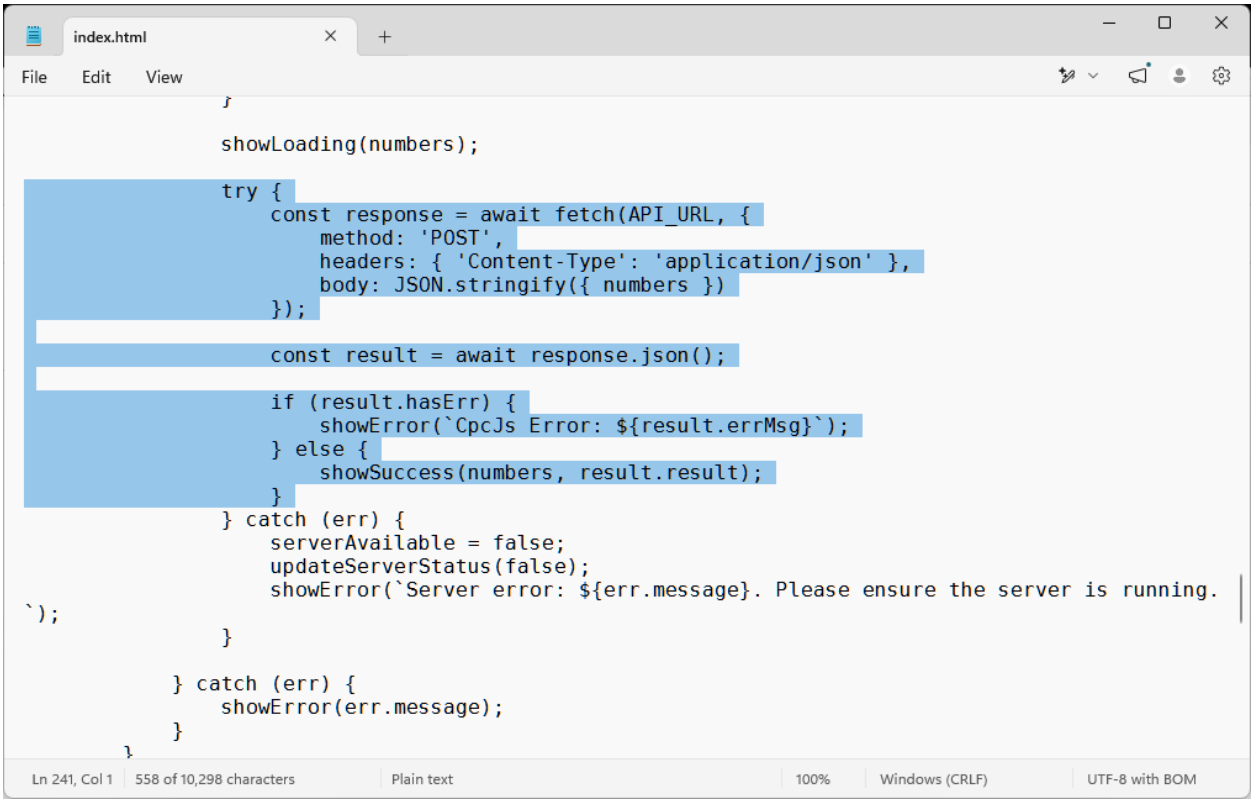
- Starts an HTTP (Node.js) server on port **3000**
- Verifies the CpcJsLibrary on server start up
- Serves the index.html web page as the default (home) page
- Exposes the POST endpoint /api/sqrts for requesting square roots of user input in the form {numbers: [1.23, 2, ...]}
- Calls **cpcJs.sqrts(numbers)** and returns the result JSON
- Handles Cross-origin Resource Sharing (CORS) between the JavaScript domain and the .Net domainHtml JavaScript web page



The C:\CpcJs\JsTestHarness\index.html file:

- Presents the main page of the application to a client browser
- Provides for user input of comma-separated, double[] vector
- Shows real-time server status
- The asynchronous function calculateSqRts() in the web page submits the user input to the Node.js wrapper in the web server
- Waits for the web server to respond with the result
- When results from the web server, the html page displays them for the client

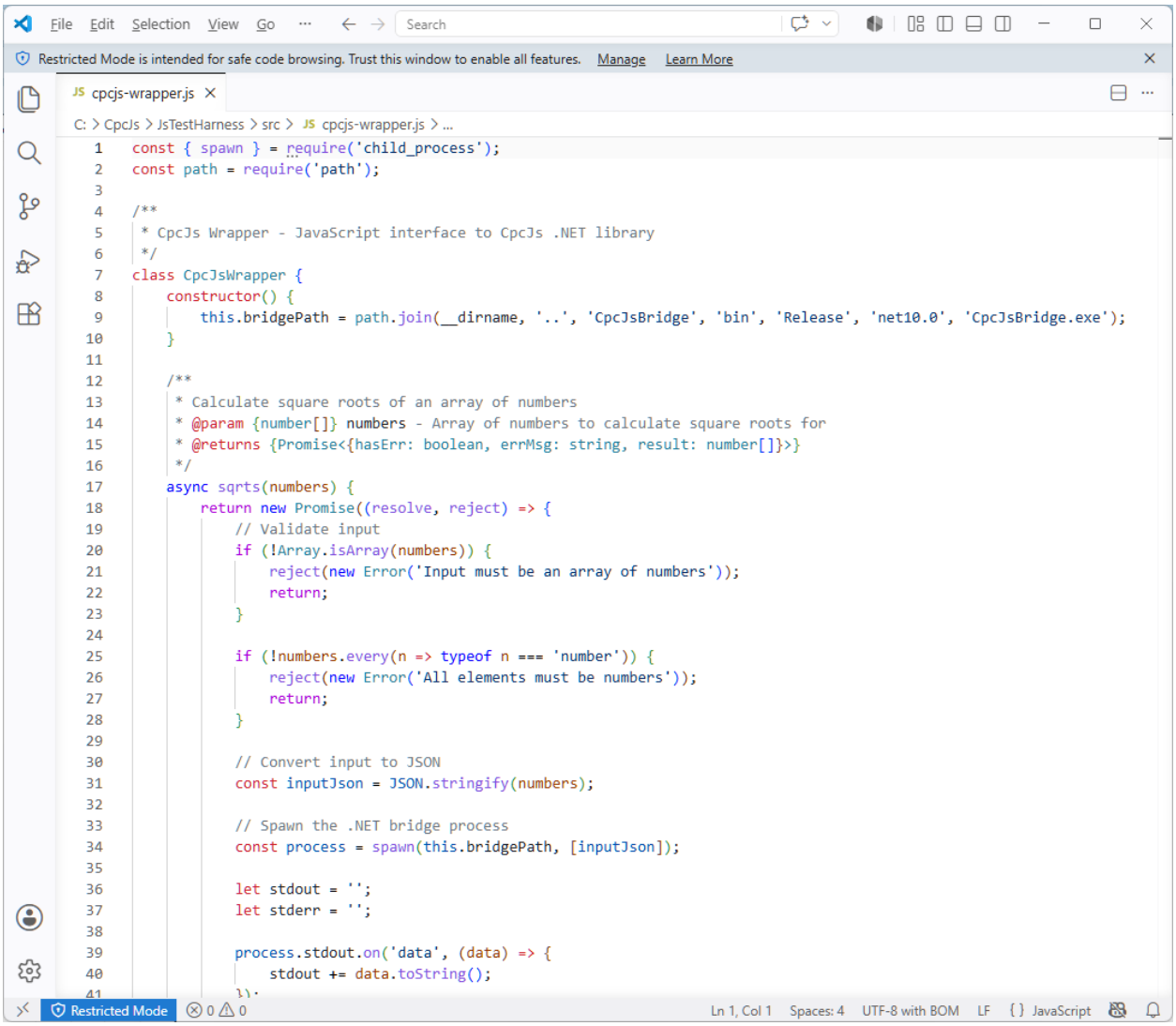
This section of the Html web page accesses the JavaScript server when the user clicks the 'Calculate Square Roots' button on the page:



Node.js JavaScript wrapper

The JavaScript wrapper: C:\CpcJs\JsTestHarness\src\cpcjs-wrapper.js file:

- Spawns the .Net 'bridge' project as a child process
- Handles json-serialization and deserialization of input and output data
- Passes the prepared input data to the .Net 'bridge' process asynchronously
- Waits for the result .Net 'bridge' process results
- Returns the result to the JavaScript web page



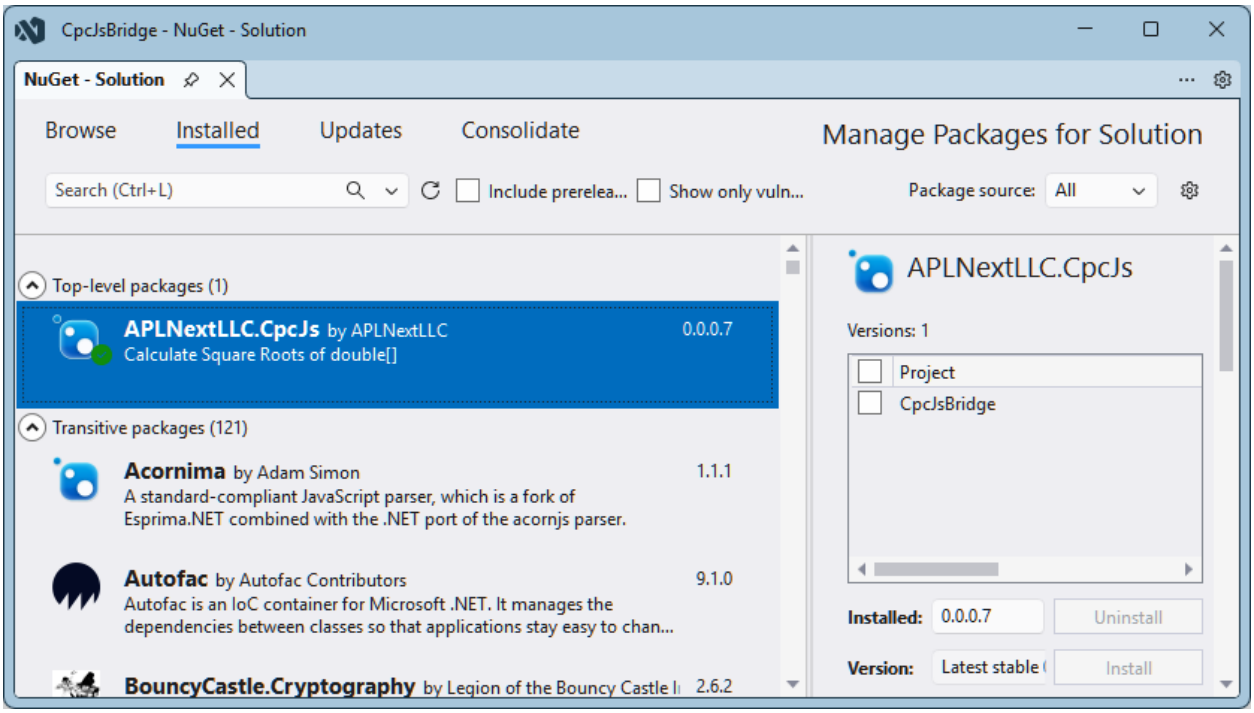
.Net Bridge Process

The .Net bridge process

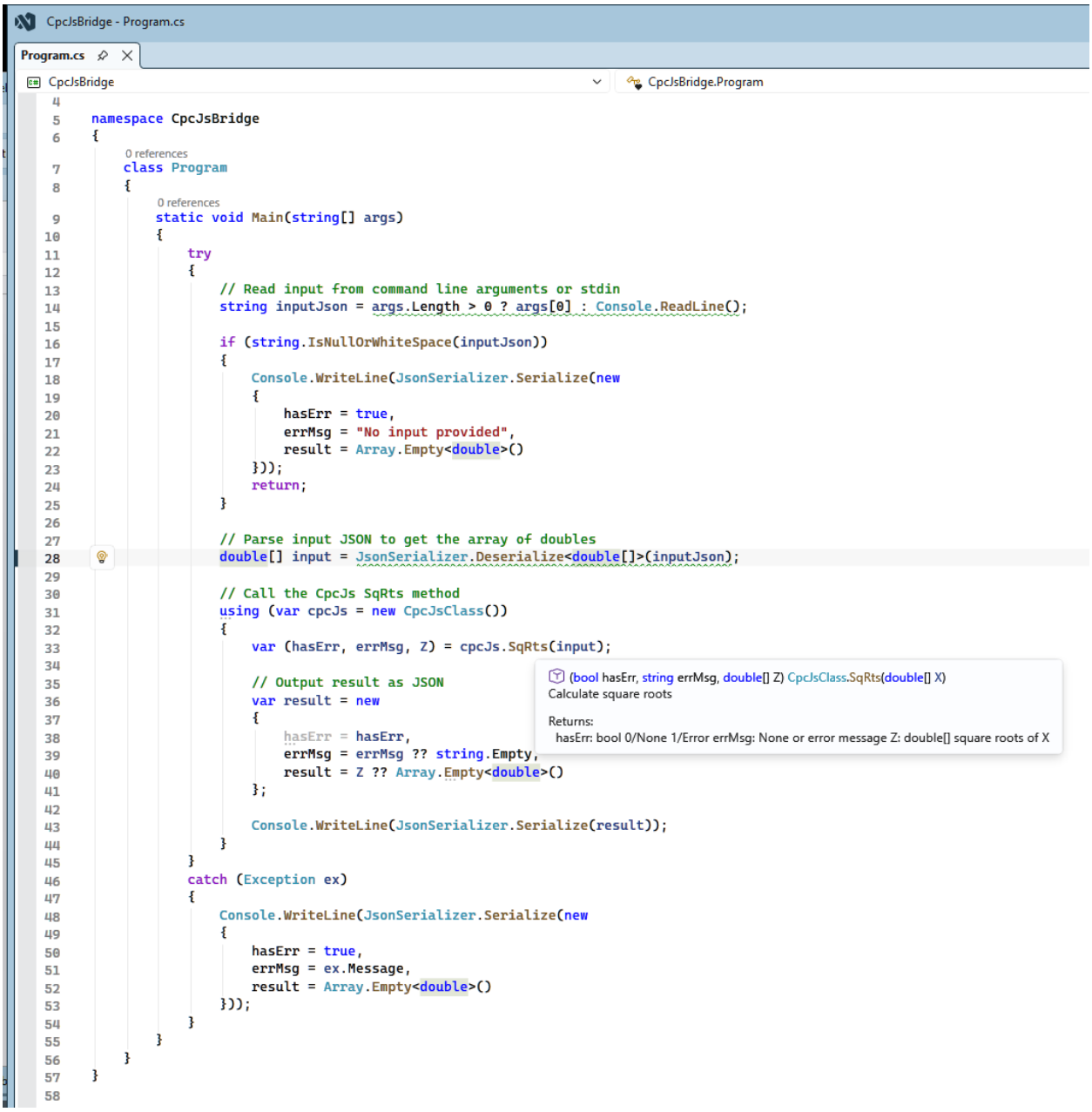
The .Net bridge process: C:\CpcJs\JsTestHarness\CpcJsBridge\Program.cs:

- Is a .Net console application
- Has a reference to the APL64 CPC Nuget package
- Accepts json-format client input via a command line argument (stdin)
- Converts the json-format input to .Net double[] data
- Calls the APL64 public CpcJsClass.SqRts() function to calculate the square roots
- Returns the result to the JavaScript wrapper

The .Net Bridge process references the APL64 Nuget package:



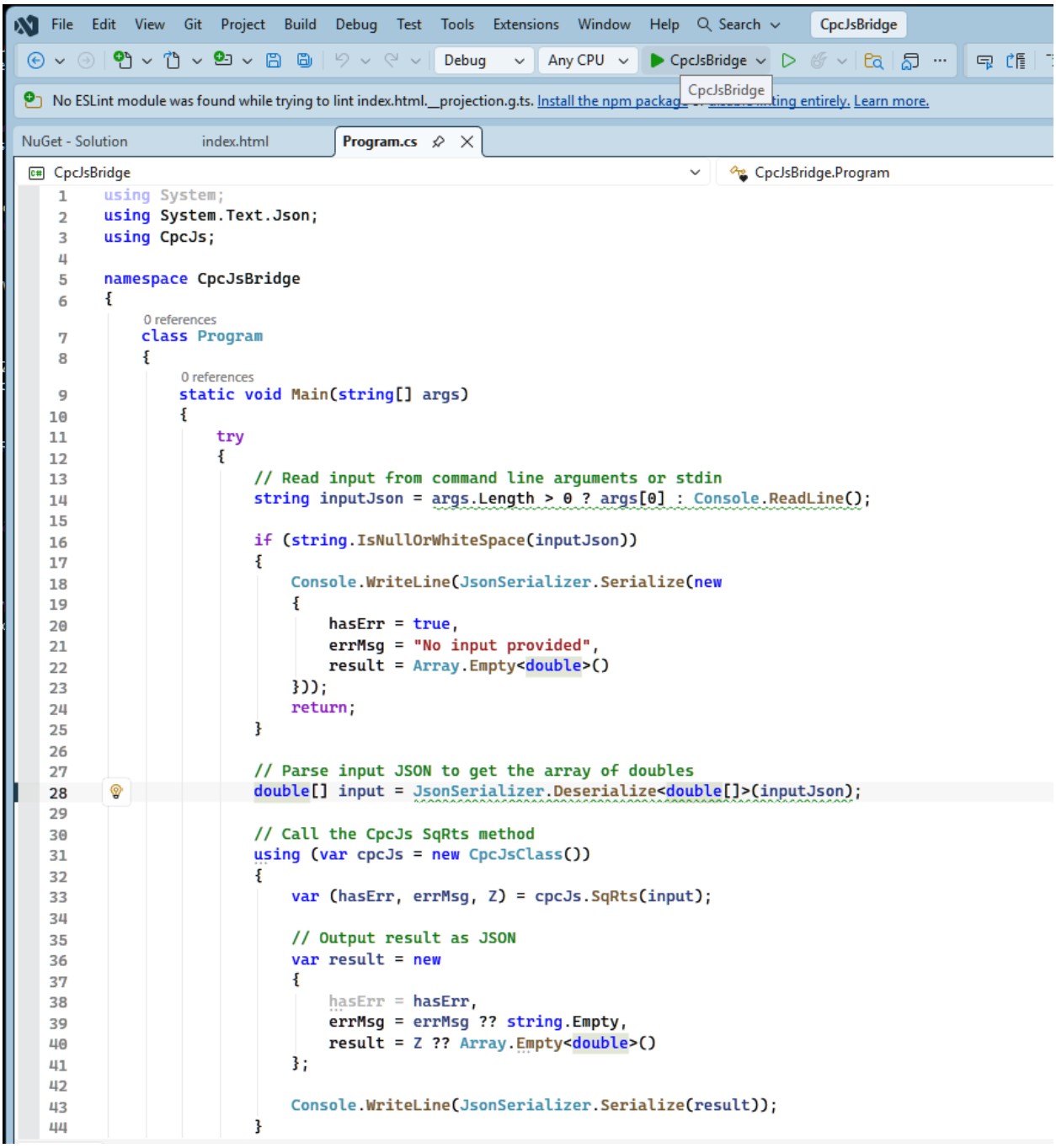
The Intellisense scaffold in the SqRts public function in the APL64 CPC provides Intellisense documentation for the end user of the APL64 CPC used in a .Net application like the .Net Bridge process:



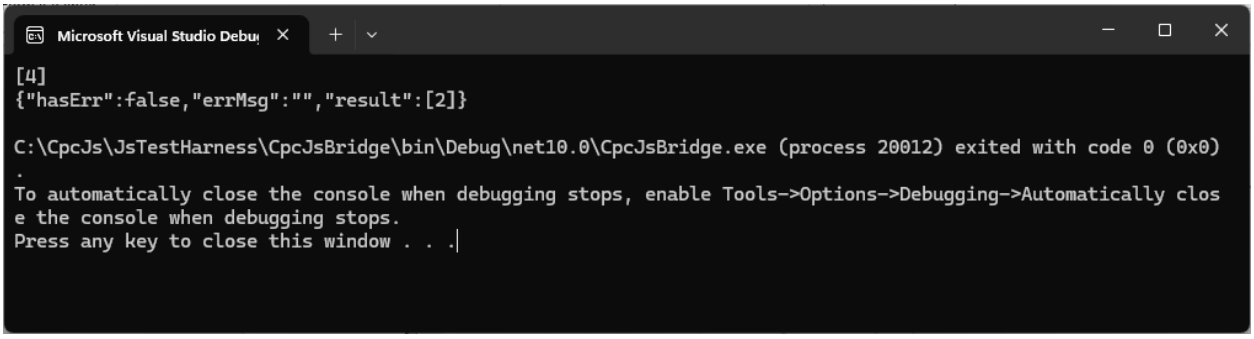

```
4
5   namespace CpcJsBridge
6   {
        0 references
7       class Program
8       {
            0 references
9           static void Main(string[] args)
10          {
11              try
12              {
13                  // Read input from command line arguments or stdin
14                  string inputJson = args.Length > 0 ? args[0] : Console.ReadLine();
15
16                  if (string.IsNullOrWhiteSpace(inputJson))
17                  {
18                      Console.WriteLine(JsonSerializer.Serialize(new
19                      {
20                          hasErr = true,
21                          errMsg = "No input provided",
22                          result = Array.Empty<double>()
23                      }));
24                      return;
25                  }
26
27                  // Parse input JSON to get the array of doubles
28                  double[] input = JsonSerializer.Deserialize<double[]>(inputJson);
29
30                  // Call the CpcJs SqRts method
31                  using (var cpcJs = new CpcJsClass())
32                  {
33                      var (hasErr, errMsg, Z) = cpcJs.SqRts(input);
34
35                      // Output result as JSON
36                      var result = new
37                      {
38                          hasErr = hasErr,
39                          errMsg = errMsg ?? string.Empty,
40                          result = Z ?? Array.Empty<double>()
41                      };
42
43                      Console.WriteLine(JsonSerializer.Serialize(result));
44                  }
45              }
46              catch (Exception ex)
47              {
48                  Console.WriteLine(JsonSerializer.Serialize(new
49                  {
50                      hasErr = true,
51                      errMsg = ex.Message,
52                      result = Array.Empty<double>()
53                  }));
54              }
55          }
56      }
57  }
58
```

The .Net Bridge Process can be tested independently of the JavaScript wrapper.

- Debug the CpcJsBridge console project to present the Console
- Enter valid json text into the Console and click Enter/Return
- The three-element result (bool hasErr, string errMsg, double[] result) will be displayed in the Console



Enter [4] to get the square root of 4:



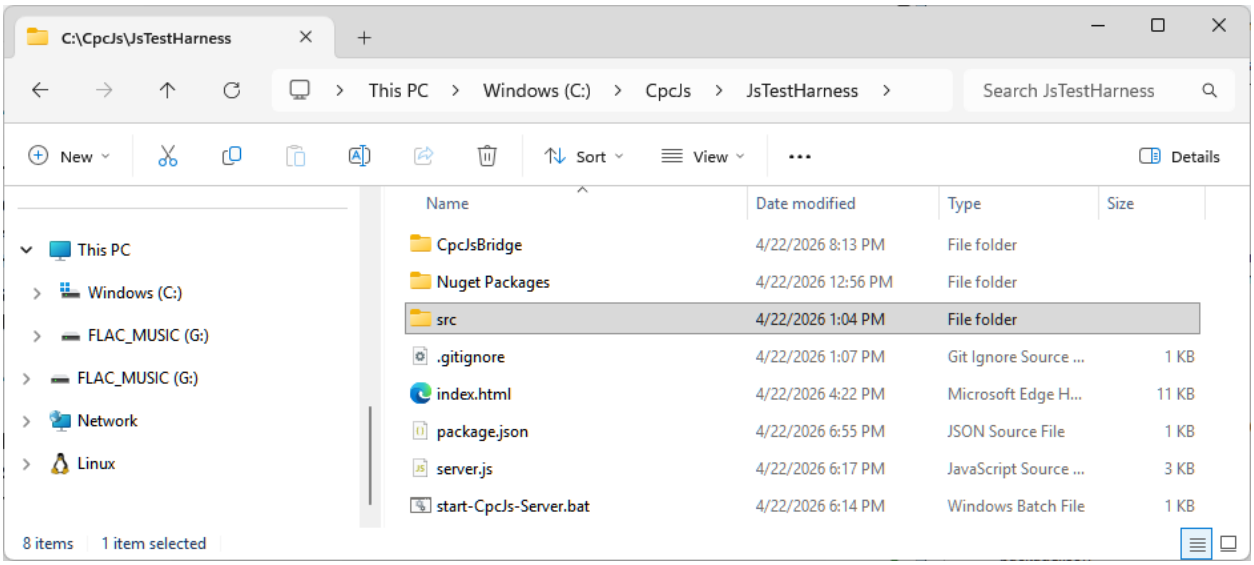
Enter [-1] to test the exception handling:

Package Configuration File

The C:\CpcJs\JsTestHarness\package.json file:

- Is the configuration file for the JavaScript project
- Defines the 'start: node server.js' script
- Includes the project metadata: name, version, description

Project File Structure



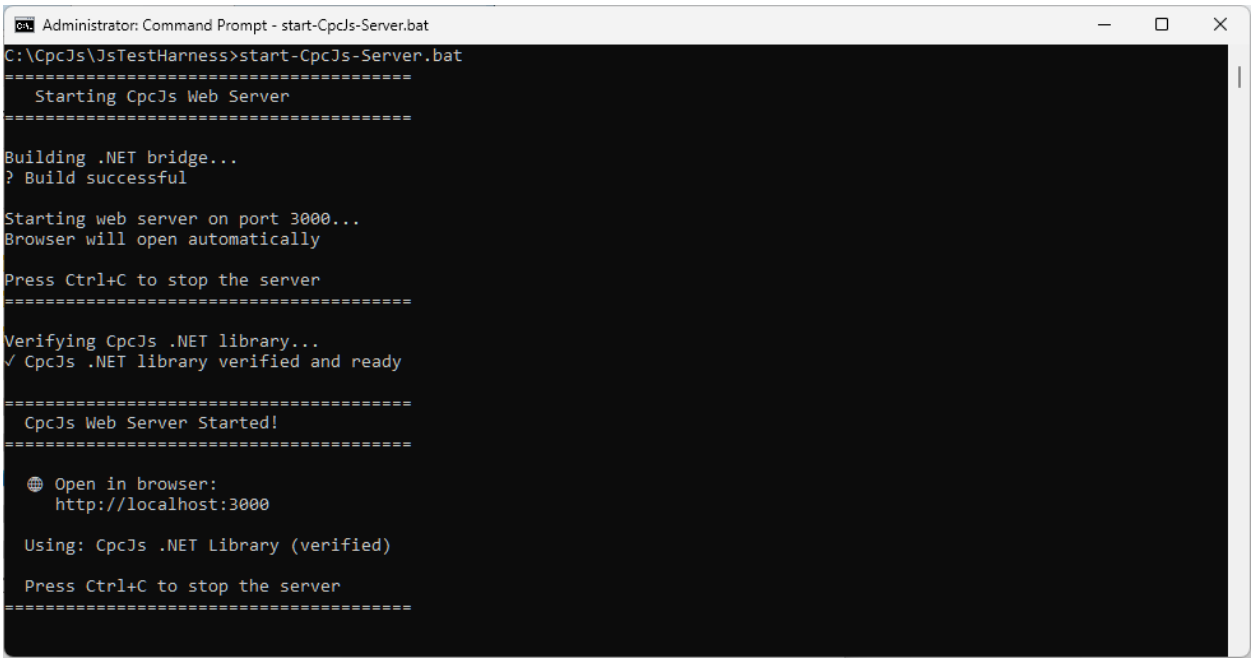
Use the JavaScript Project

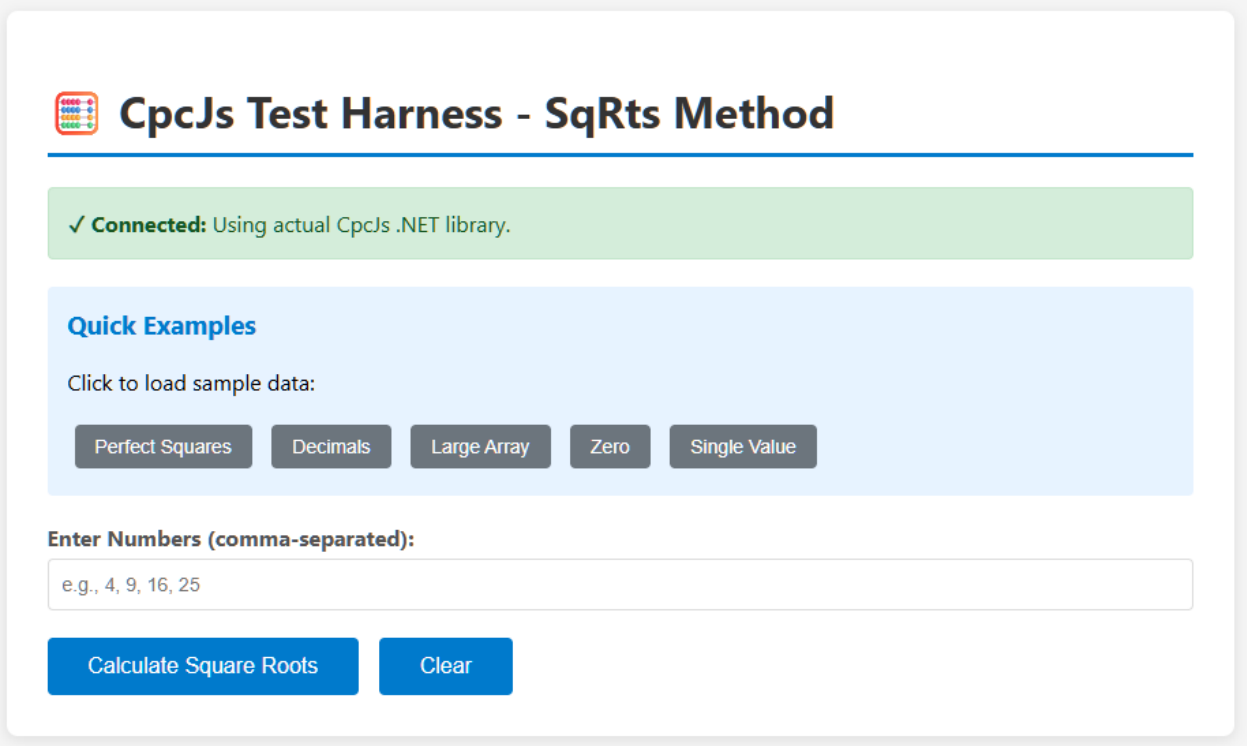
Start the web server

Using a Command Prompt as administrator, run these commands:

- cd C:\CpcJs\JsTestHarness
- start-CpcJs-Server.bat

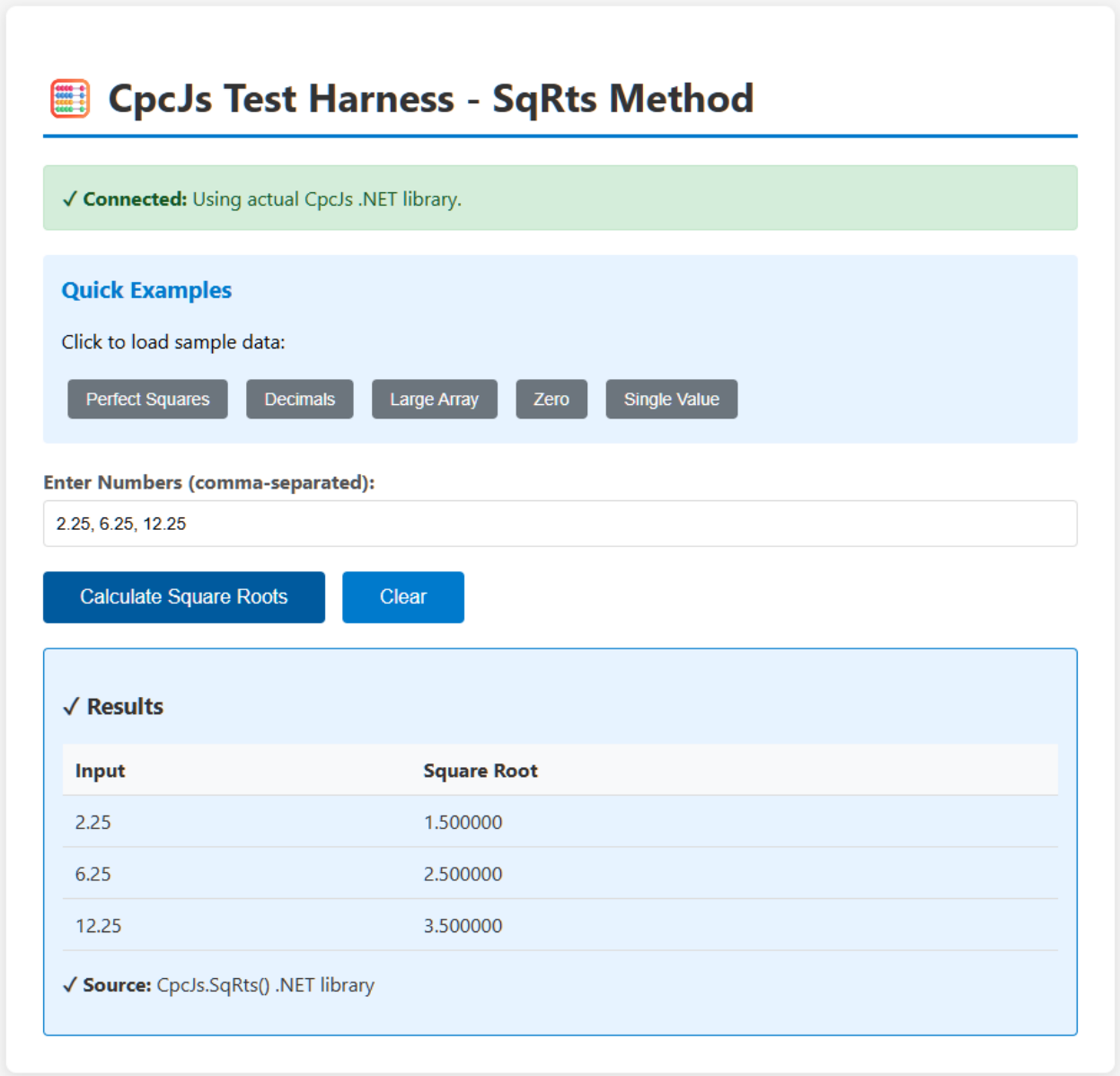
The web server will start, and the home page will be presented in the client's default browser at the localhost:3000 (CpcJs Test Harness) url:





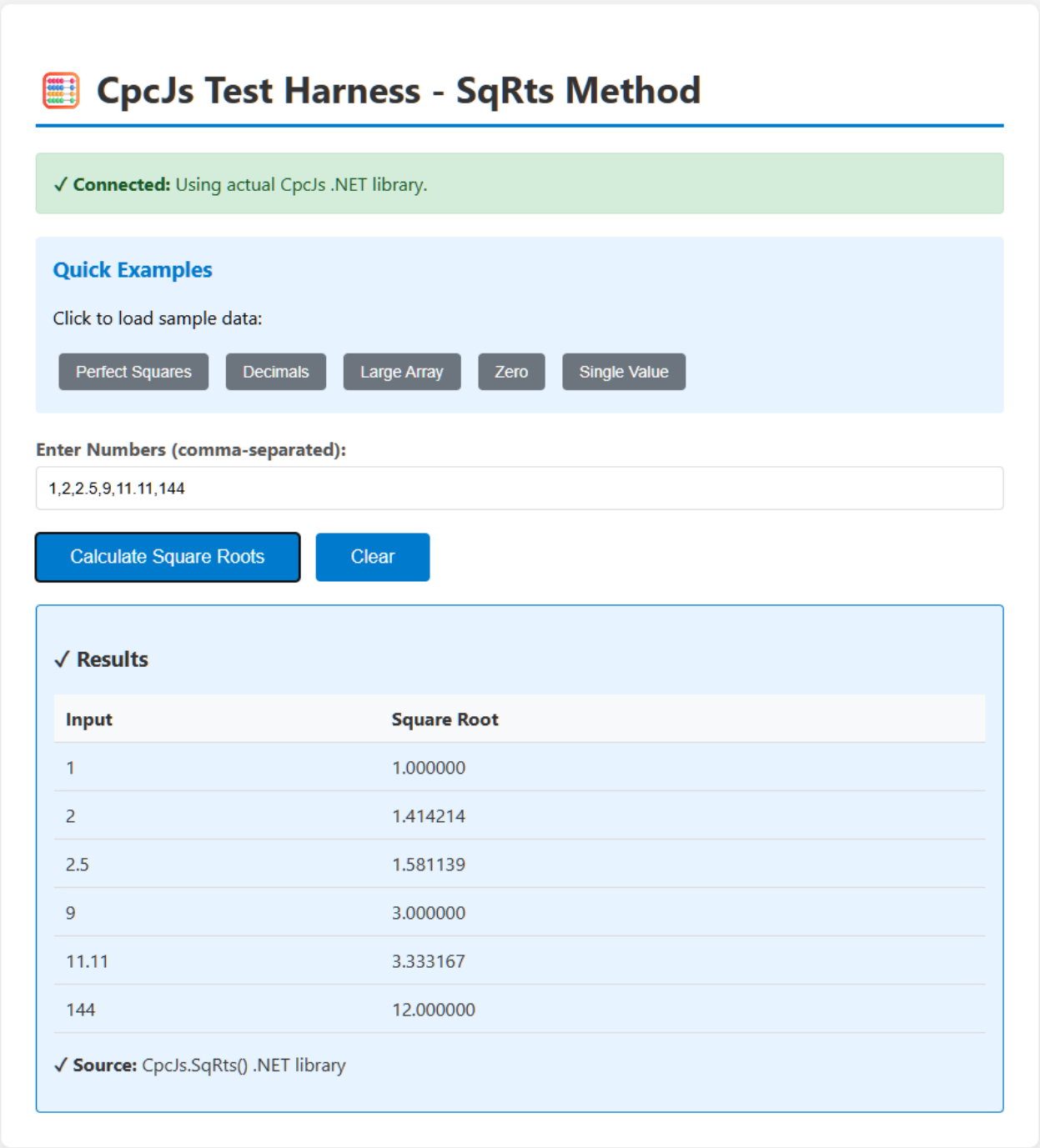
Try a quick example

- Click the ‘Decimals’ button
- Click the ‘Calculate Square Roots’ button



Enter user input and get results

- Enter some comma-separated numbers
- Click the ‘Calculate Square Roots’ button



Test the Exception Handling

The exception message is provided by the public function in the APL64 CPC.

Enter -2 and click Calculate Square Roots: