

Using a VisualAPL Cielo Script in a .Net Application System Solution

Suppose that the APLNext VisualAPL Cielo Explorer has been used to investigate and design a component of an application system and this design work has been tested in a Cielo Explorer 'script'.

This document describes two ways to use the VisualAPL Cielo Explorer script in a .Net application system solution:

A. VisualAPL Class Library Derived from the Cielo Explorer Script

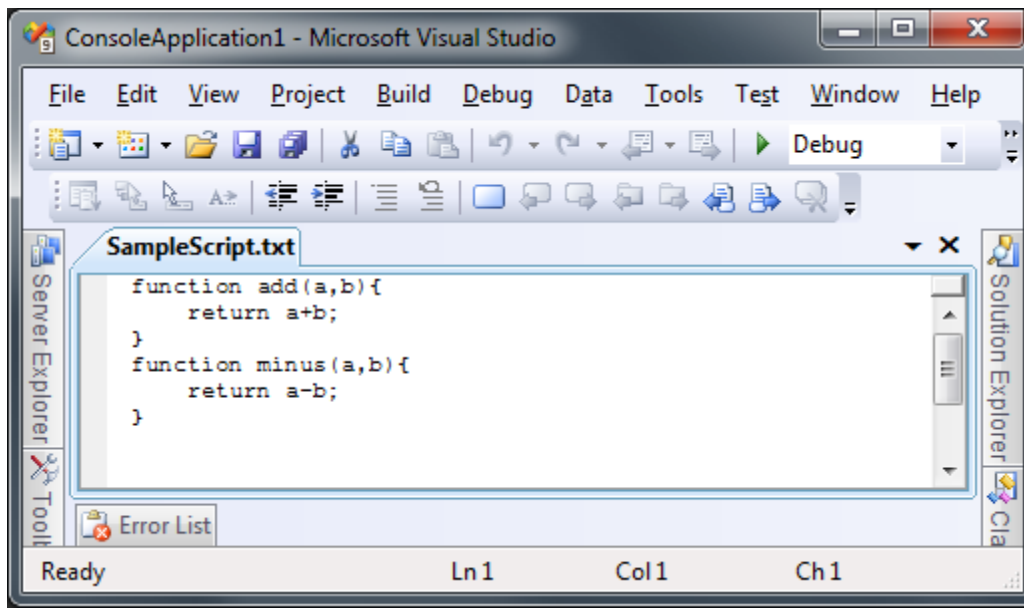
The recommended way to integrate the VisualAPL source code in the Cielo Explorer script is to use it as the basis for a VisualAPL .Net assembly (class library or .dll) project which will then be referenced by another project in the application system solution.

This method will provide fully-managed code and make the VisualAPL class library easy for a .Net programmer to integrate into the application system solution. A C# or VB.Net project can reference and directly use the methods, properties and events in the VisualAPL class library just like those in any other .Net assembly in a language-independent manner. The arguments to the VisualAPL class library methods are created within the scope of the calling environment. The Visual Studio debugging facility will transparently handle errors and warnings, if any, resulting from the VisualAPL class library.

Only the public methods and properties defined in the VisualAPL class library will be available to the environment which references this class library.

The Cielo Explorer script serves as a test platform to perfect the application algorithms to be implemented in VisualAPL.

1. Create and perfect the Cielo Explorer script and save it as part of the solution as 'SampleScript.txt' so that it can be used to test the VisualAPL class library functions. In the example provided, the Cielo Explorer script contains two VisualAPL functions, 'add' and 'minus', each with two arguments. The .Net method syntax is used for these because they will be called by the C# environment.



2. Create the C# 'ConsoleApplication0' project in the Visual Studio 2008 solution. This console application will serve as the program environment that will reference the VisualAPL class library and call the methods in that library.

The programming environment referencing the VisualAPL class library need not be a console project, but can be any .Net project (.exe or .dll), such as WPF GUI project, C# class library, C# WCF windows service, VB.Net project, VisualAPL project. etc.

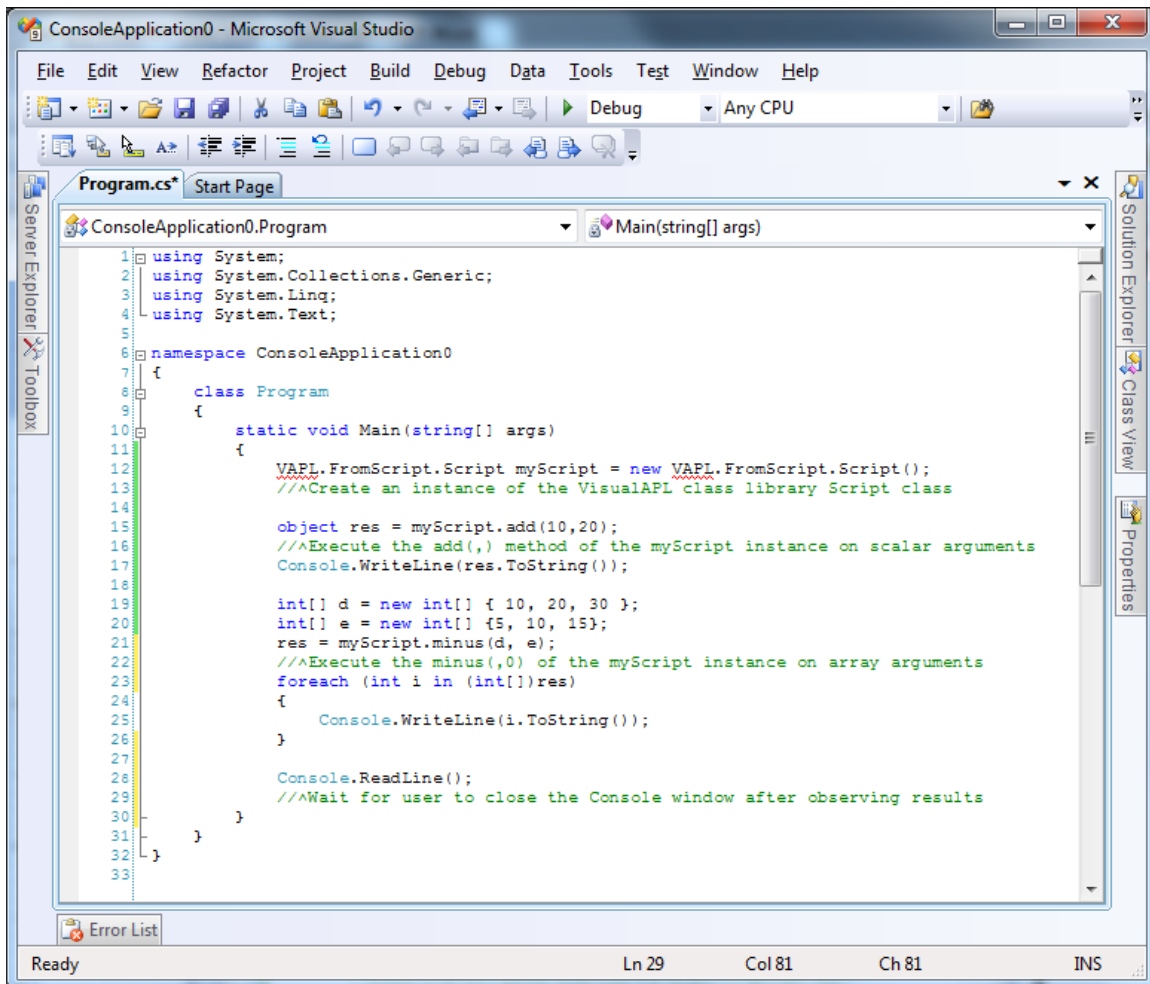
3. Add a Visual Studio VisualAPL class library project to the solution with the following source code.

```
using System;
using System.Collections.Generic;
using System.Text;
using APLNext;
namespace VAPL.FromScript
{
    public class Script
    {
        public function add(a,b)
        {
            return a+b;
        }

        public function minus(a,b)
        {
            return a - b;
        }
    }
}
```

The 'public' keyword is used so that these methods will be exposed to the C# project referencing this VisualAPL class library.

4. In the C# console project add a reference to the VisualAPL class library project.
5. Add the C# source code to the C# console project to use the VisualAPL class library methods.



6. Debug and run the solution to see the results.

Other examples of this VisualAPL class library methodology are found here:

<http://forum.apl2000.com/viewtopic.php?t=478>
<http://forum.apl2000.com/viewtopic.php?t=543>
<http://forum.apl2000.com/viewtopic.php?t=225>

B. Compile and Execute the Cielo Explorer Script from C#

The 'APLEngine' included with the VisualAPL product can be used to compile and execute a Unicode text string which contains the Cielo Explorer script.

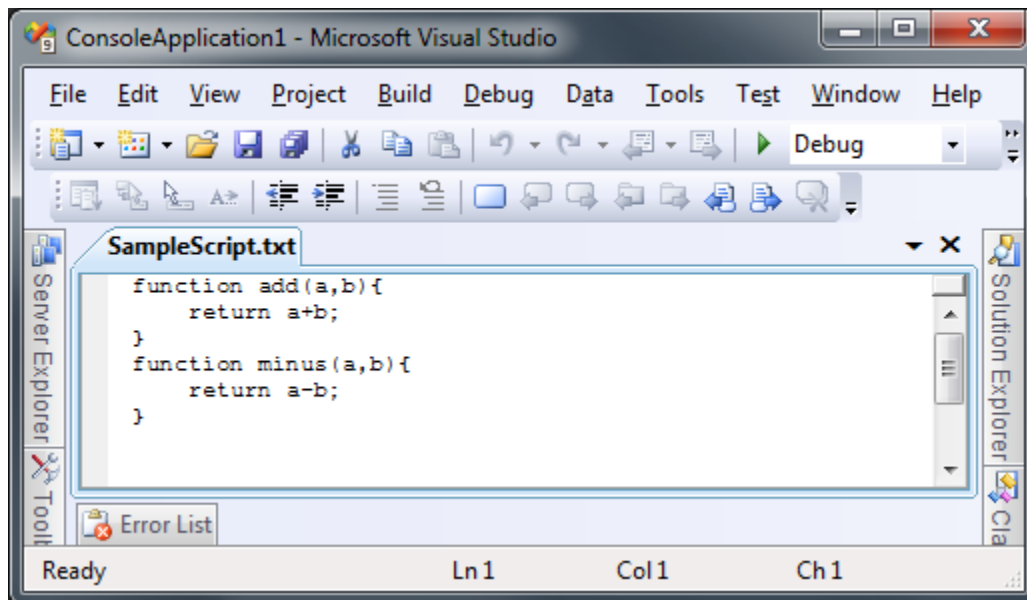
Since the VisualAPL source code is not contained in a .Net assembly, this method requires that a VisualAPL-specific interface be used by the calling environment (e.g. C#) that will use the Cielo Explorer script.

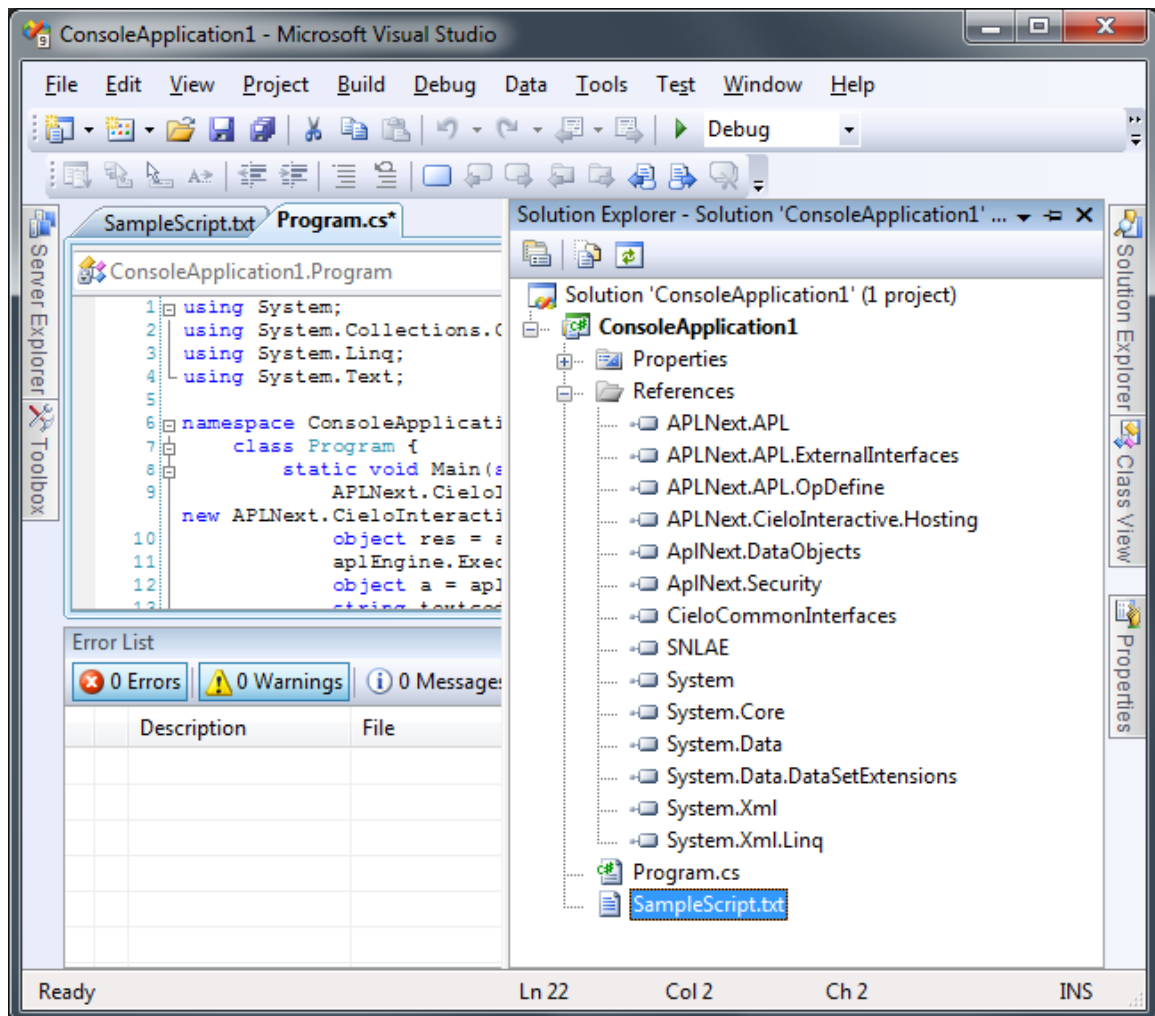
Errors and warnings, if any, in the Cielo Explorer script will not be presented in a conventional .Net manner. This condition requires that the Cielo Explorer script be thoroughly debugged prior to compiling and executing it.

The syntax to evaluate a method in the Cielo Explorer script involves creating a text string representing the method name and arguments. The method arguments are created within the instance of the compiled and executed Cielo Explorer script rather than in the scope of the calling environment.

Associated with this document is a Visual Studio 2008 solution which illustrates this methodology for a simple C# console project using the Cielo Explorer script.

1. Create and perfect the Cielo Explorer script and save it as part of the solution as 'SampleScript.txt'. In the example provided, the Cielo Explorer script contains two VisualAPL functions, 'add' and 'minus', each with two arguments. The .Net method syntax is used for these because they will be called by the C# environment.





It is of course possible to create much more elaborate Cielo Explorer scripts and such scripts can reference .Net assemblies for more advanced functionality. Some examples are provided here:

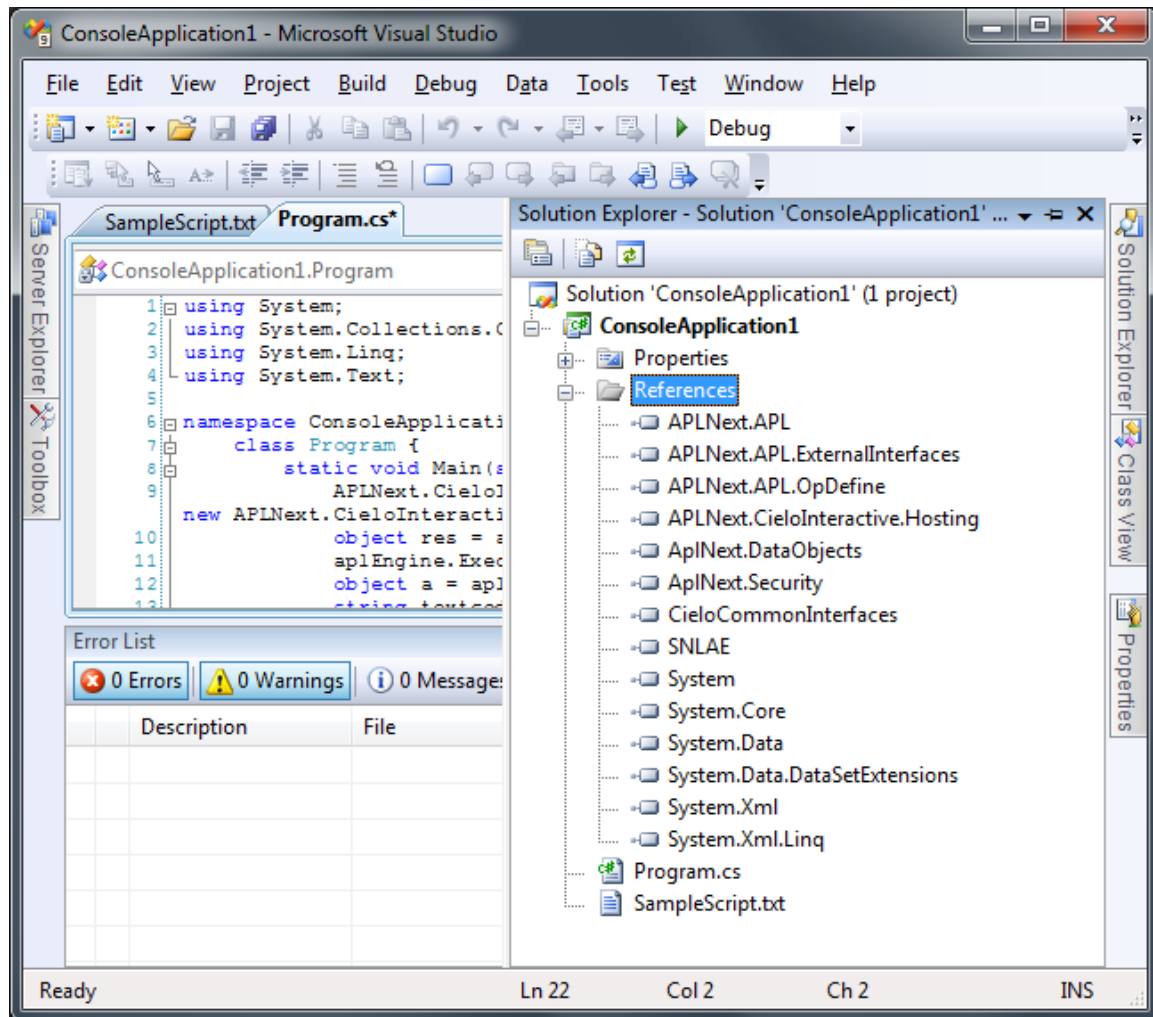
<http://forum.apl2000.com/viewtopic.php?t=486>
<http://forum.apl2000.com/viewtopic.php?t=485>
<http://forum.apl2000.com/viewtopic.php?t=489>
<http://forum.apl2000.com/viewtopic.php?t=474>

2. Create the C# 'ConsoleApplication1' project in the Visual Studio 2008 solution. This console application will serve as the program environment that will read in the Cielo Explorer script as Unicode text and then compile and execute it using the 'APLEngine'.

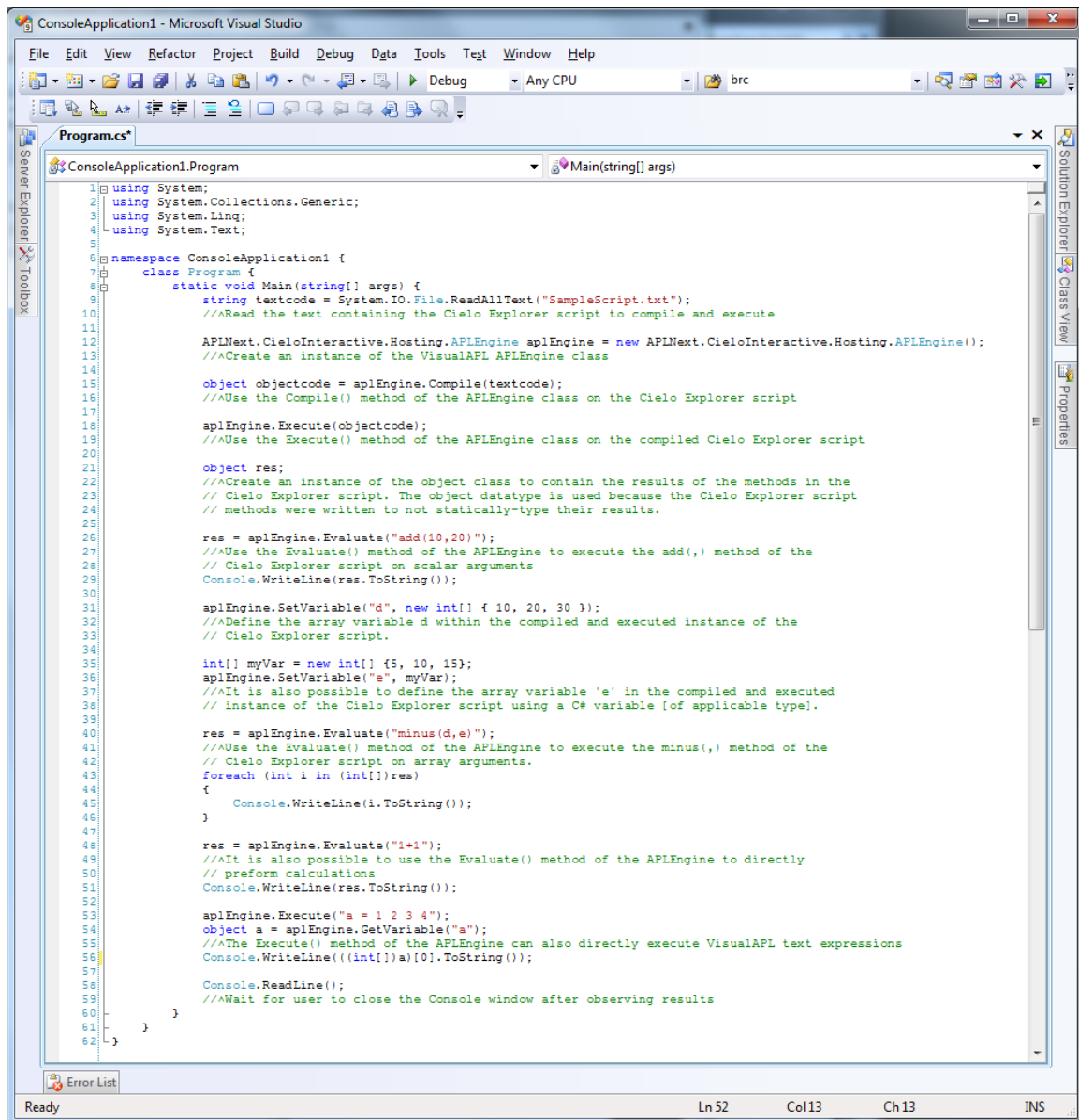
The programming environment using the Cielo Explorer script need not be a console project, but can be any .Net project (.exe or .dll), such as WPF GUI

project, C# class library, C# WCF windows service, VB.Net project, VisualAPL project, etc.

3. Add the appropriate VisualAPL references to the C# 'ConsoleApplication1' project. These are found in the directory in which VisualAPL was installed.



4. Add the C# source code to the C# console project to read the Cielo Explorer script text file from the solution directory, compile it, execute it and call methods implemented within it:



The VisualAPL APL engine instance created in the C# source code has an `Execute()` method that can be used to execute text strings representing VisualAPL or C# statements.

5. VisualAPL Cielo Explorer scripts can contain immediate execution statements in addition to implementations of methods. These immediate execution statements are those VisualAPL or C# statements contained in the Cielo Explorer script that are not in a method definition. When the compiled object code derived from the Cielo Explorer script is executed using the `Execute()` method of the VisualAPL APL engine, these immediate execution statements, if any, are immediately executed.
6. Debug and run the solution to see the results