OpenSpan Article: Runtime Hosting

Published: April 1, 2008

Runtime Hosting Overview

OpenSpan enables you to launch and host the OpenSpan Runtime from within your own .NET applications. The OpenSpan Runtime is the execution engine required to run OpenSpan solutions.

OpenSpan Runtime Hosting Step by Step

The following step by step process shows you how to host the OpenSpan Runtime in a .NET application (.NET versions 2.0 or newer).

1. Create a RuntimeHost class instance. The runtime class exists in the OpenSpan.Runtime namespace within the OpenSpan.Runtime assembly.

   There are several overloads of the runtime constructor available, and the most commonly used are the following:

   - Runtime ()
   - Runtime (IServiceProvider *serviceProvider*)
   - Runtime (IServiceProvider *serviceProvider*, string *sessionName*)

     The *serviceProvider* should be passed when the .NET hosting application provides its own custom services required by components that exist in the loaded OpenSpan project.

     SessionName is needed when multiple OpenSpan runtime instances are running in the same .NET application. The *sessionName* uniquely identifies each runtime instance so that the debugger can attach to the separate instances.

2. Set the RuntimeHost.ProjectPath to the full path of the OpenSpan project (*.proj) or deployment (*.openspan) file.

3. Use the Runtime.Start() method to launch specified project. The RuntimeHost.Started event will fire after the project is successfully loaded.

4. Stop the project and unload the Runtime instance by calling the RuntimeHost.UnloadRuntime() static method. Note that you must pass the runtime instance into the UnloadRuntime() method as a reference parameter. The RuntimeHost.Stopped event will fire when the project stops.

   Note: An OpenSpan solution named SampleSolution is used in the code examples that follow. The SampleSolution deployment file is attached to this PDF document.

Sample Code

```
// Create an instance of the OpenSpan Runtime, passing in the global ServiceContainer
mRuntimeHost = new RuntimeHost(mServiceContainer);

// Listen for notification that the project has started
mRuntimeHost.Started += new ContextEventHandler(RuntimeHost_Started);

// Listen for notification that the project has stopped
mRuntimeHost.Stopped += new EventHandler(RuntimeHost_Stopped);

// Load the SampleSolution from the application directory
mRuntimeHost.ProjectPath = @”SampleSolution.OpenSpan”;

// Start the OpenSpan Runtime
mRuntimeHost.Start();
```

Adding a Custom Service

Follow these steps to expose custom service functionality inside of OpenSpan.

1. Add a service to the service provider that is created with the RuntimeHost.

Sample Code

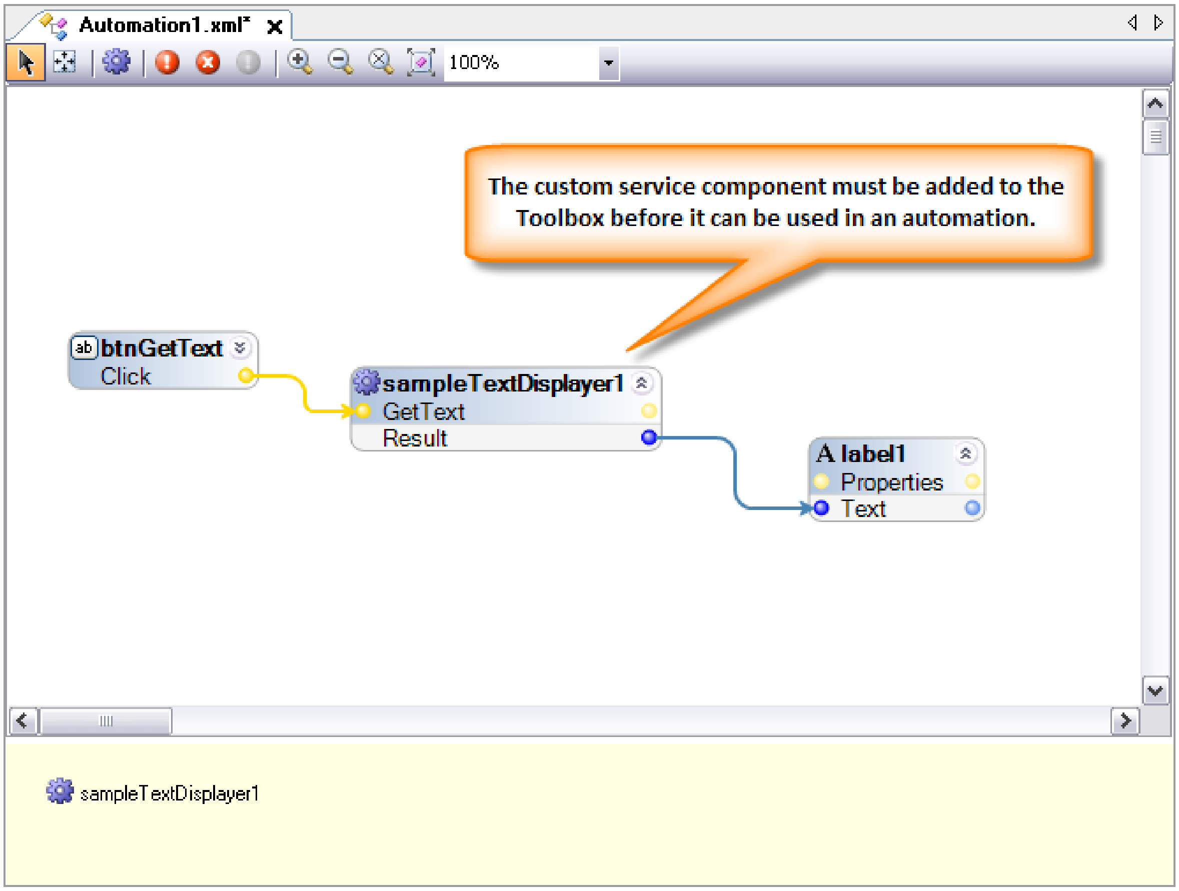
```
ServiceContainer serviceContainer = new ServiceContainer();

serviceContainer.Add(typeof(ISampleService), this); // where ‘this’ implements ISampleService

RuntimeHost runtimeHost = new RuntimeHost(serviceContainer);
```

2. Create a component (that is, inherit from Component or implement IComponent) that can exercise methods and properties of the custom service. Components are selected in the OpenSpan Studio **Toolbox** window (more on that below).

3. As methods and properties are exercised the component will require access to the custom service. A reference to this service can be retrieved from the component's site.

Sample Code
ISampleService sampleService = this.Site.GetService(typeof(ISampleService) as ISampleService;

The following automation is part of the sample solution provided with this document and it demonstrates the use of a custom service component.



Running the Sample Solution and Host Application

Use the sample OpenSpan solution and .NET application attached to this document to execute an OpenSpan Runtime instance from a .NET application.

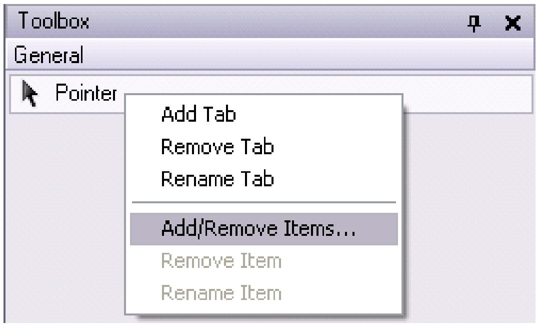
1. Using the *SampleService.zip* file attached to this PDF document, extract the *SampleHost.exe* and *SampleService.dll* files to OpenSpan Studio program folder.

   SampleHost.exe is a small .NET application and *SampleService.dll* is the custom service used in the sample solution automation.

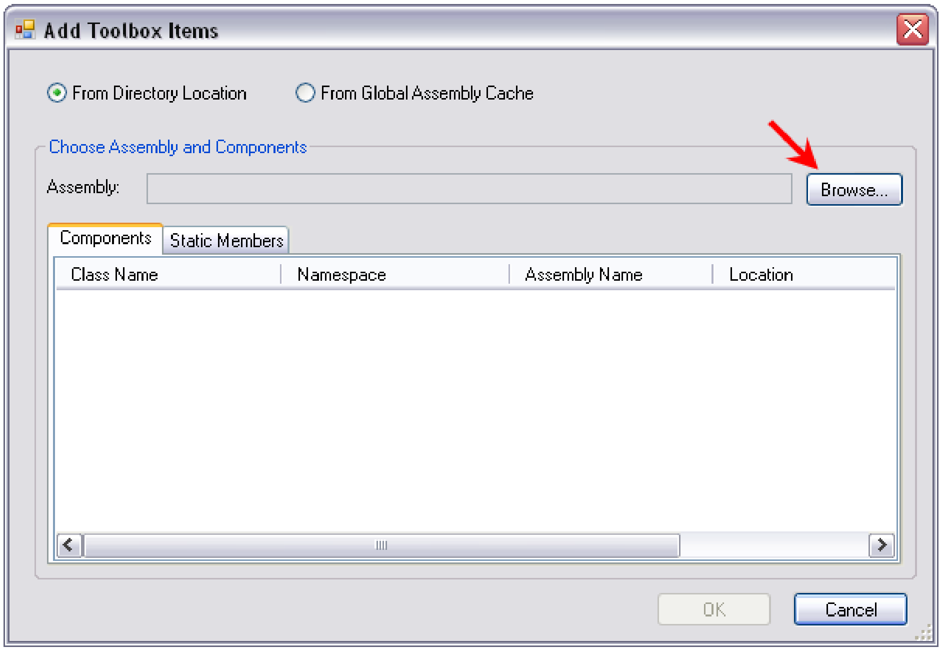
2. Save the *SampleSolution.OpenSpan* deployment file, also attached to this PDF document, to the OpenSpan Studio folder. Then create the solution contained in the deployment file in a sub-folder of the OpenSpan Studio Program folder.

   Note: This solution works only with OpenSpan Studio versions 3.2 or newer.

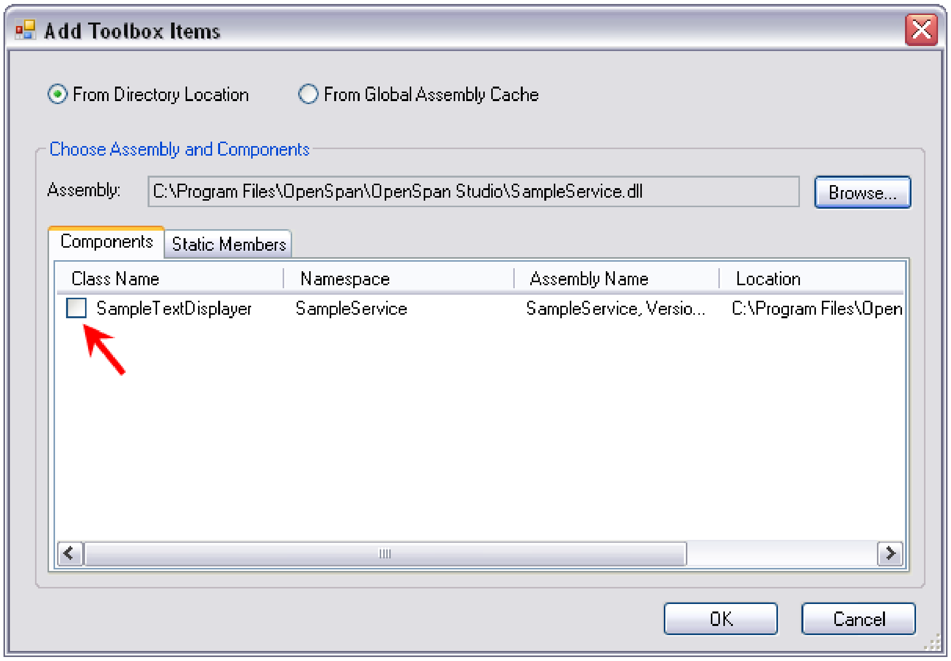
3. Add the *SampleTextDisplayer* component to the OpenSpan Toolbox. The *SampleTextDisplayer* is the custom service used in the sample automation. **Right-click** within any tab on the **Toolbox** window and select **Add/Remove Items** from the context menu.



4. The **Add Toolbox Items** dialog opens. Browse to the custom component's assembly.



5. Check the box next to the component's Class Name and click OK. The component now appears in the **Toolbox** window and is available for use in automations.


Add Toolbox Items
● From Directory Location
○ From Global Assembly Cache
Choose Assembly and Components
Browse...
Components
Static Members
OK
Cancel

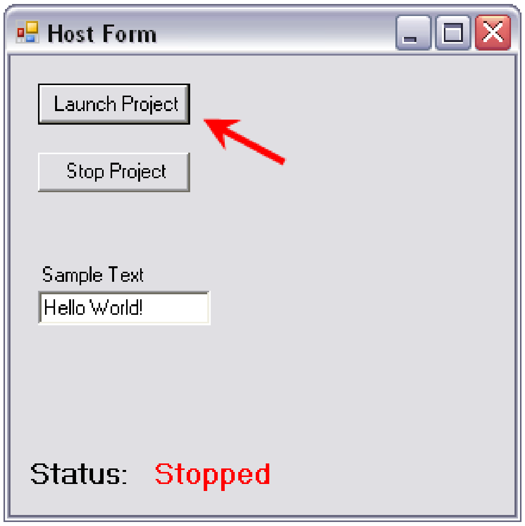
Assembly: C:\Program Files\OpenSpan\OpenSpan Studio\SampleService.dll

Class Name	Namespace	Assembly Name	Location
□ SampleTextDisplayer	SampleService	SampleService, Versio...	C:\Program Files\Open

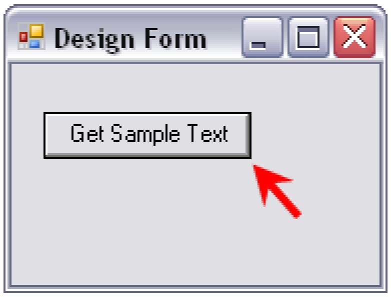
6. Review both the Windows form and automation contained in the solution, but don't run the solution.

7. Close OpenSpan Studio.

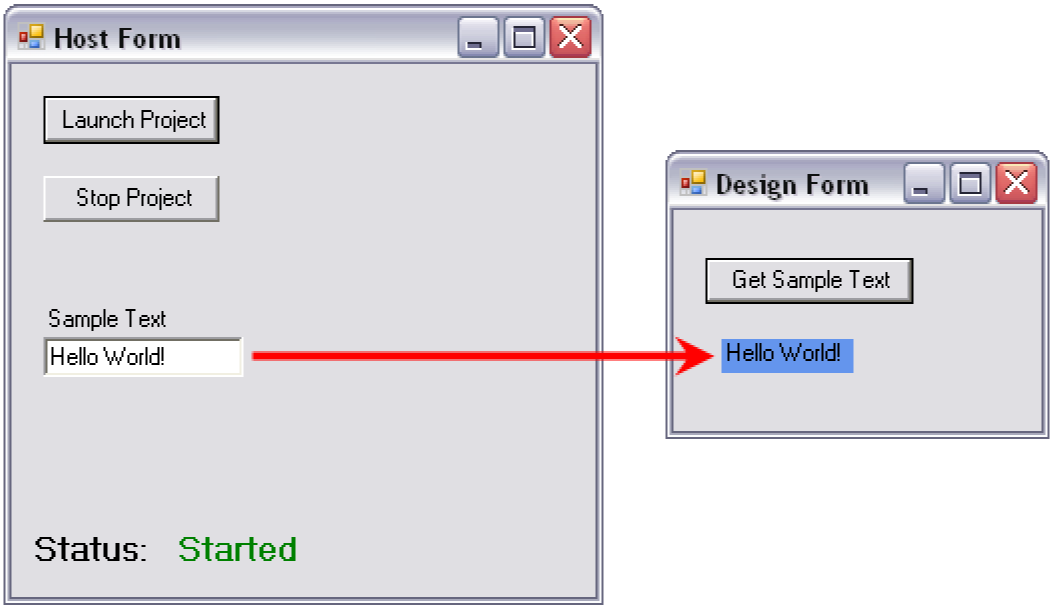
8. Execute the *SampleHost.exe* .NET application that you extracted to the OpenSpan Studio program folder. The **Host Form** window opens.



9. Select the **Launch Project** command button in the **Host Form** window, which launches the OpenSpan Runtime and the *SampleSolution* project. The **Design Form** Windows form opens.



10. Click the **Get Sample Text** command button to run the OpenSpan automation contained in the sample solution. The sample solution copies the contents of the **Sample Text** text box on the .NET application **Host Form** to the **Design Form** window that is part of the OpenSpan solution.



Runtime Hosting Benefits

Runtime hosting grants you full control over an OpenSpan Runtime instance for the duration of its execution, allowing you to start and stop it based on your requirements. Additionally, Runtime hosting gives you access to custom functionality available in your code, extending standard OpenSpan functionality.

More Resources

For more information about .NET applications, see the many articles in the Microsoft® MSDN® developer program at www.msdn.microsoft.com.

OpenSpan Studio Help documentation

To access material within OpenSpan Studio, choose **Help** | **Help Contents**, select the **Search** tab, and search on the keywords *Runtime* and *Hosting.*

About the Author

Danny McDougald is OpenSpan's Director of Product Development.