



PureWeb[®] STK 4.0

Quick Start Guide: C# Service

The information contained herein is proprietary and confidential and cannot be disclosed or duplicated without the prior written consent of Calgary Scientific Inc.

Copyright © 2013 Calgary Scientific Inc. All rights reserved.

About Calgary Scientific

Calgary Scientific Inc., is dedicated to providing advanced visualization, web enablement, mobility enhancement solutions to industries looking for secure access and use of their data or graphics intensive applications, while using their existing systems. Visit www.calgaryscientific.com for more information.

Notice

Although reasonable effort is made to ensure that the information in this document is complete and accurate at the time of release, Calgary Scientific Inc. cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in this document may be incorporated in future versions.

Your Responsibility for Your System's Security

You are responsible for the security of your system. Product administration to prevent unauthorized use is your responsibility. Your system administrator should read all documents provided with this product to fully understand the features available that reduce your risk of incurring charges for unlicensed use of Calgary Scientific products.

Trademarks

© 2013 Calgary Scientific Inc., ResolutionMD, PureWeb and the Calgary Scientific logo are trademarks and/or registered trademarks of Calgary Scientific Inc. or its subsidiaries. Any third-party company names and products are for identification purposes only and may be trademarks of their respective owners.

Released by

Calgary Scientific Inc. www.calgaryscientific.com.

Document Version: PW4.0_CSharpService_QSG_07-2013_v1.000.00

Table of Contents

Preface	4
Chapter 1	Introduction.....	5
Chapter 2	Deployment Procedures	6
	Prerequisites.....	6
	Setting Up the Environment.....	7
	Setting the Environment Variables.....	7
	Installing .NET 3.5 on Microsoft Windows Server 2012.....	8
	Installing PureWeb	9
	Building the Sample Application	9
Chapter 3	Running the Scribble Application.....	10
	Reloading the Plugins.....	10
	Launching the Application	10
	Exercising the Sample Functionality.....	11
	Diagnostics Panel	11
	Collaboration.....	12
Chapter 4	Exploring the Code.....	13
	Scribble Service Code Pointers.....	13
	Program.cs File.....	13
	ScribbleControl.cs File	14
	RemotedControl.cs File	14
Chapter 5	Debugging.....	15
	Procedure	15

Preface

Welcome to the *PureWeb STK 4.0 Quick Start Guide: C# Service*, a document that provides instructions to install the PureWeb STK, and to build, run and troubleshoot sample PureWeb C# service applications.

Intended Audience

This document is primarily intended for software developers who plan to install and use the STK to PureWeb enable a C# service. It has been written with the assumption that you have:

- a working knowledge of Microsoft Visual Studio,
- an intermediate knowledge of C#,
- a basic knowledge of Silverlight, Java Swing, Flex, iOS or HTML5 (for client development).

Making Comments on This Document

If you especially like or dislike anything about this document, feel free to e-mail your comments to techpubs@calgaryscientific.com.

Contacting Calgary Scientific Support

Use one of the methods in the table below to contact Calgary Scientific support.

Web Site	E-Mail
support.getpureweb.com	support@getpureweb.com

1 Introduction

PureWeb[®] is a platform that enables the rapid transformation of enterprise software into cloud-ready, web and mobile applications.

The PureWeb STK (software transformation kit) is a set of tools that provides developers everything they need to implement the PureWeb solution in their own applications, in particular service and client APIs, documentation and API reference material, and sample applications in each supported programming language.

The sample applications illustrate key concepts of PureWeb enablement; they can be used by developers as a starting point to create their own PureWeb-enabled applications.

The PureWeb *Quick Start Guide* series is intended to get developers up and running as quickly as possible, using these sample applications as models.

As its name indicates, the current document, *PureWeb STK Quick Start Guide: C# Service*, focuses on the C# sample service. The guide provides instructions to install the PureWeb STK, and to build, run and troubleshoot a sample service.

Because the PureWeb enablement of a service is usually done in tandem with the creation of a client, the sample C# service comes with a sample Silverlight client, whose functionality is also described in this document. If you prefer to work with a client other than Silverlight, please refer to the *Quick Start Guide* for your preferred client development environment.

Recommended Reading

This *Quick Start Guide* is intended as an introduction, and does not provide in-depth information. It should be used in conjunction with the other guides of the PureWeb STK documentation suite, in particular: *Installation Guide*, *Developer's Guide*, *Server Administration Guide*, and C# service API reference material.

To access these references in a Windows operating system, press the Windows key on the keyboard to display the **Start** menu (**Start** screen in Windows 8), then type "pureweb" as the search term.

2 Deployment Procedures

In this chapter, you will find prerequisite information as well as instructions to install the PureWeb STK and build the sample C# service.

Because a service application is normally developed in tandem with a client application, the instructions below will also build a Silverlight sample client. If you prefer to work with one of the other sample clients, please refer to the *Quick Start Guide* for that client environment.

For more detailed instructions, such as how to handle a side-by-side installation of the STK, please refer to the complete *PureWeb STK Installation Guide*.

Prerequisites

To install the PureWeb STK, you will need:

- Java Development Kit (JDK) 1.7
- C++ Runtime, either 2008 SP1 ATL x86/x64 or 2010 SP1 x86/x64; you can download the 2010 version from the following location:
www.microsoft.com/en-us/download/details.aspx?id=5555
- A Windows 64-bit operating system, either of: Windows 7, Windows 8, Windows Server 2008 R2, or Windows Server 2012

Note: Although only a 64-bit platform is supported for application development, you can target the applications developed using the PureWeb STK to deploy on either a 32-bit or a 64-bit platform.

To build and run the C# sample service application (including the Silverlight client), you will need:

- .NET 3.5
- Microsoft Visual Studio, either of: 2008 SP1 or 2010 SP2
- Microsoft Silverlight v5.0 Tools for Visual Studio 2010 and Microsoft Silverlight 5 Toolkit (Silverlight 5 does not work with Visual Studio 2008).

Note: You can develop PureWeb applications using .NET 4.0, but the sample applications require .NET 3.5.

Microsoft Windows Server 2012 comes with .NET 4.0 by default. To work with .NET 3.5 on this operating system, you will have to install it separately. For instructions, see “Installing .NET 3.5 on Microsoft Windows Server 2012” on [page 8](#).

Setting Up the Environment

The PureWeb server requires the `JAVA_HOME` and `Path` environment variable to be set. Also, if working in Microsoft Windows Server 2012, you may need to install .Net 3.5 manually. Procedures for completing these operations are described in this section.

Setting the Environment Variables

On a Windows operating system, to access the Environment Variables dialog:

1. Press the `Windows` key on your keyboard to display the **Start** menu (or **Start** screen in Windows 8).
2. Type “environment variables” to search for that string.
3. Select the option **Edit the System Environment Variables** from the search results. This will display the System Properties dialog box’s Advanced tab.
4. Click the **Environment Variables** button.

JAVA_HOME

Check that you have a `JAVA_HOME` entry under System variables:

- If the entry is there and the path points to version 1.7 of the JDK, skip this section and check your `Path` variable, as described in next section.
- If the entry is there, but the path points to the wrong version of the JDK: click the **Edit** button, change the path in the **Variable value** field and click **OK**.
- If the entry is missing:
 - a. Click the **New** button.
 - b. Type `JAVA_HOME` in the **Variable name** field.

- c. Enter the path to JDK 1.7 in the **Variable value** field (for example, the default for update 6 is `C:\Program Files\Java\jdk1.7.0_6`)
- d. Click the **OK** button to save your changes.

Path

If your `Path` system variable is missing the path to the JDK:

1. In the list of system variables, select the `Path` variable.
2. Click on the **Edit...** button.
3. Add the following to the **Variable value** string: `%JAVA_HOME%\bin;`

Click the **OK** button to save your changes

Note: After changing environment variables, if you had a console window or Visual Studio open, you will need to restart it for these changes to take effect.

Installing .NET 3.5 on Microsoft Windows Server 2012

The procedure below is required only if you want to develop using .NET 3.5 on a Microsoft Windows Server 2012 operating system, or if you want to build and run the C# sample service applications.

.NET 3.5 is not installed by default on Microsoft Windows Server 2012. To install it, you will need the Windows Server 2012 .iso file or DVD.

1. Insert the .iso file or DVD into a drive.
2. Open Server Manager and select **Add Roles and Features**.
3. Click **Next** in the wizard until you reach the Features page.
4. Select **.NET Framework 3.5 Features** and click **Next** to display the Confirm installation selections screen.
5. Select the **Specify an alternate source path** link.
6. Enter the following as the path, changing the directory if needed:
`D:\Sources\SxS`
7. Click **OK**.
8. Click the **Install** button.

Installing PureWeb

If you already have PureWeb installed, uninstall it first.

To install the PureWeb STK in a Windows operating system:

1. Double-click on its executable file, `PureWeb_Win_Setup-[version].exe`, and follow the instructions in the wizard. The installation directory path cannot contain any spaces.
2. Install your PureWeb server license, a `.lic` file which you will have received from Calgary Scientific separately from the STK:
 - a. Navigate to `[Installed_directory]\Server\conf\`.
 - b. Place the `.lic` license file into that folder.
 - c. Double-click on the **Start PureWeb** desktop icon to start the server.
3. Check that the following environment variables are set correctly:
 - ♦ `PUREWEB_HOME:`
`[Installed_directory]\Server`
 - ♦ `PUREWEB_LIBS:`
`[Installed_directory]\SDK\Redistributable\Libs`

Building the Sample Application

To build Scribble in Microsoft Visual Studio:

1. Navigate to:
`[Installed_directory]\SDK\Samples\Scribble\ScribbleAppCSharp`
2. Double-click on the Microsoft Visual Studio solution file:
`ScribbleApp2010.sln`.
3. In Microsoft Visual Studio, Go to the **Build** menu, then select **Build Solution**.
4. Check the output to ensure that the build was successful.
5. Check that the service files can be found at this location:
`[Installed_directory]\Server\bin\ScribbleApp`

3 Running the Scribble Application

This section contains instructions on how to run the sample application, Scribble, and describes the functionality available in this application.

Note: The default server and port for PureWeb is `localhost:8080`. Replace as appropriate if this is not correct for your implementation.

Reloading the Plugins

This step is only necessary the first time you build the service.

1. If the PureWeb server is not currently running, launch it by double-clicking on the **Start PureWeb** desktop icon.
2. Open a browser and navigate to:
`http://localhost:8080/pureweb/config/plugins`
3. Enter `admin` in both the **Name** and **Password** fields.
4. Click the **Sign In** button to open the Configuration page.
5. Click **Reload Plugins**, under the PureWeb server Configuration heading.

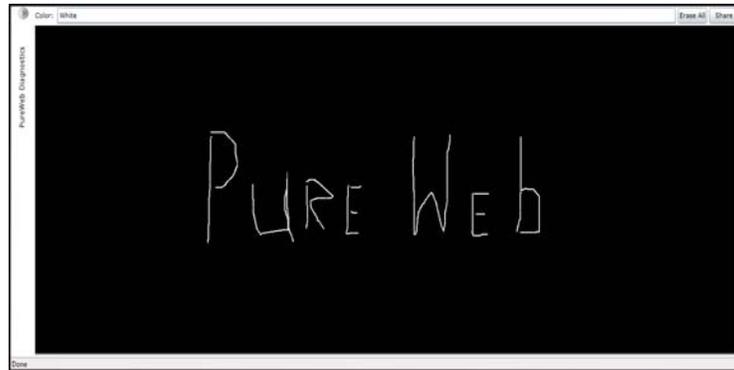
Launching the Application

1. If the PureWeb server is not currently running, launch it by double-clicking on the **Start PureWeb** desktop icon.
2. Open a browser window and navigate to the client's URL:
`http://localhost:8080/app?name=ScribbleApp&client=silverlight&_diagnostics=true`

The `_diagnostics=true` parameter is optional. See “Diagnostics Panel” on [page 11](#).

Exercising the Sample Functionality

The Scribble sample application is a simple canvas; users can draw lines by left-clicking the mouse and dragging.



In addition, Scribble offers the following functionality:

- Change colors by entering the name of a new color in the **Color** text field.
- Clear the canvas by clicking on the **Erase All** button.
- Share the application with other users by clicking the **Share** button (for more information, see the Collaboration section, on next page).
- Display the Diagnostics Panel by clicking on the small arrow button located in the upper-left corner of the client window (for more information, see the Diagnostics Panel section, below).

Notice how the service window and client windows are in sync:

- When you resize the client browser window, the C# service window resizes to the same size.
- When you close the browser window or tab containing the client, the service application window also closes.

Diagnostics Panel

The Diagnostics Panel debugging tool is described in detail in the Debugging chapter of the *PureWeb STK Developer Guide*, however below is a brief overview of the functionality it offers.

The panel is composed of four tabs: Options, Trace, App State, and Bandwidth.

- The **Options** tab is used to define options that impact image bandwidth and quality.
- The **Trace** tab is used to examine the messages to and from the server. The controls allow you to set the behavior of the trace view, including the buffer length, scrolling, and clearing of the buffer.
- The **App State** tab displays an XML representation of the application's state.
- The **Bandwidth** tab is used to measure latency and bandwidth.

Collaboration

The Scribble sample contains functionality to illustrate collaboration. This section describes how one user can share the application with a second user so that they are both logged in the same session.

A few important points to note before getting started:

- By default the Scribble client connects using the hostname of the local host. If you are trying to share the application from another computer, you must modify the URL, `http://localhost:8080/pureweb/...`, by replacing `localhost` with the IP address or DNS name.
- Cookie sharing between browser tabs may cause issues with the functionality of the sample application. We recommend that you use two separate browser instances if you want to open two sessions of the same application. Chrome users can also use an incognito window.

First user:

1. Click the **Share** button located in the upper-right corner of the Scribble client application.
2. Copy the URL displayed in the popup window and close the window.
3. Provide the URL you copied to the person you want to share the session with.

Second user:

1. Open a browser and navigate to the shared URL to display the **PureWeb Collaboration Login** page.
2. Select a Preferred Client from the drop-down box. All possible clients are listed, but only the ones actually built will work.
3. Enter `Scientific` as your password and click the **Sign In** button to join the session.

4 Exploring the Code

One of the main purposes of including sample applications with the STK is for developers who are new to PureWeb to use these samples as a starting point for coding and to experiment with them.

Therefore, don't be shy about editing the code and trying new things. That's the best way to learn, and you'll see how quickly you'll become proficient with PureWeb.

This chapter gives some pointers about the code in the sample C# service; however, it assumes that you are familiar PureWeb concepts such as views, state, commands and event handlers, which are described in the *Developer's Guide*.

For an overview of the client-side code, please refer to the *Quick Start Guide* for each of the supported client programming languages.

Scribble Service Code Pointers

To inspect the Scribble application code described in this section, double-click on the `ScribbleApp2010.sln` file located at:

```
[Installed_directory]\SDK\Samples\Scribble\ScribbleApp
```

Program.cs File

This file contains the `main` method, used to create an instance of the `StateManager` class, which is responsible for managing application state, responding to input events and commands sent from the client, generating responses such as updated images to send back to the client application

The `main` method also creates an instance of `StateManagerServer`, which implements an IO thread that receives the input events and commands from the client via the PureWeb server. The input events and commands are dispatched to the `StateManager` for processing. Responses are gathered from the `StateManager` and returned to the client via the PureWeb server.

Finally, the `main` method also starts the server.

ScribbleControl.cs File

The `ScribbleControl.cs` file contains the `ScribbleControl` class, which implements the business logic of the Scribble service application. It contains a number of methods and inner classes that are noteworthy from a PureWeb enablement perspective.

OnLoad

This method performs a number of important actions including:

- registering the `ScribbleView` instance with the state manager,
- adding an application state property change listener to listen for color changes,
- adding a command handler to listen for erase commands from the client,
- initializing the `ScribbleColor` application state property to the color `WHITE`.

OnScribbleColorChanged

This method responds to changes to the `ScribbleColor` application state property. Whenever the client application changes `ScribbleColor`, a state difference is generated and transmitted to the service application. The `StateManager` merges the difference into the application state on the server-side, and notifies any registered event handlers of the change.

OnExecuteClear

This method respond to `Clear` commands sent by the client application.

RemotedControl.cs File

This file contains the `RemotedControl` class, a user control that implements remote rendering through the PureWeb `IRenderedView` and `IRemoteRenderer` interfaces. This class contains two public methods: `RemoteRender()` and `RemoteRenderImmediate()`.

The `RemoteRender()` method is used to defer the PureWeb server from obtaining an updated image for the client until such time as an image response is about to be sent.

5 Debugging

Using the sample application as models, this chapter describes how you would debug a PureWeb C# service using Microsoft Visual Studio 2010; the procedure may be slightly different if you are working with a different version.

Other sources of debugging information include the Diagnostics Panel section of the Debugging chapter in the *Developer's Guide*, as well as the Debugging chapter in each of the *Quick Start Guide* for client development.

Procedure

1. Open `ScribbleApp2010.sln` in Microsoft Visual Studio.
2. Build the solution from the **Build** menu or by pressing `Ctrl+Alt+F7`.
3. If the PureWeb server is not running, launch it by double-clicking the **Start PureWeb** desktop icon.
4. Open a browser window (Microsoft Internet Explorer).
5. Navigate to the following URL to launch the application:
`http://localhost:8080/app?name=ScribbleApp&client=silverlight`
6. When prompted for login information, type `admin` in both the **Name** and **Password** fields, then click the **Sign In** button.
7. Select **Debug | Attach To Process** in Microsoft Visual Studio.
8. Select `ScribbleApp.exe` from the list of Available Processes and click the **Attach** button.
9. Set a breakpoint in the `OnExecuteClear` method in the `ScribbleControl.cs` file.
10. Click the **Erase All** button in the client application to trigger the breakpoint.