PureWeb® STK 4.0

Quick Start Guide: Silverlight Client

About Calgary Scientific

Calgary Scientific Inc. is dedicated to providing advanced visualization, web enablement, and 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

Released by

Calgary Scientific Inc. www.calgaryscientific.com.

Document Version: PW4.0_SilverlightClient_QSG_07-2013_v1.000.00

Table of Contents

Preface

Welcome to the *PureWeb STK 4.0 Quick Start Guide: Silverlight Client*, a document that provides instructions to install the PureWeb STK, and to build, run and troubleshoot sample PureWeb Silverlight client applications.

Intended Audience

This document is primarily intended for software developers who plan to install and use the STK to PureWeb enable a Silverlight client. It has been written with the assumption that you have a working knowledge of:

• Microsoft Visual Studio
• Microsoft Silverlight

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: Silverlight Client*, focuses on the Silverlight client. The guide provides instructions to install the PureWeb STK, and to build, run and troubleshoot a sample client.

Two sample Silverlight clients are available:

- Scribble allows users to draw lines on a canvas, change line color, and erase lines.

- Asteroids is a PureWeb-enabled implementation of the classic video game with a twist -- it supports both single-player and two-player modes; each player independently controls his or her own ship.

Before working with the sample Silverlight client, a sample service is needed; instructions for building the sample service so can be found in the *Quick Start Guide* for each service development platform.

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 other PureWeb reference material, in particular:

• *Installation Guide*

• *Developer's Guide*

• *Server Administration Guide*

• Silverlight 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 a sample Silverlight client.

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 x86/x64 or 2010 SP1 ATL x86/x64; the 2010 version can be downloaded from the following location:
  www.microsoft.com/en-us/download/details.aspx?id=5555
- Windows 64-bit operating system, either of: Windows 7, Windows 8, Windows Server 2008 R2 or Windows Server 2012

Note: Although only 64-bit platforms are 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 Silverlight sample client application (either Scribble or Asteroids), you will need:

- .NET 3.5
- Microsoft Visual Studio 2010 SP2
- Microsoft Silverlight v5.0 Tools for Visual Studio 2010 SP1
  www.microsoft.com/en-us/download/details.aspx?id=28358
- Microsoft Silverlight 5 Toolkit
  silverlight.codeplex.com/releases/view/78435

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 .

Setting Up the Environment

PureWeb requires the `JAVA_HOME` and `Path` environment variables. Also, if working in Microsoft Windows Server 2012, you may need to install .NET 3.5 manually. The procedures for 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 that string.

3. Select **Edit the System Environment Variables** from the search results. This will display the System Properties dialog box, with the Advanced tab selected.

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.
  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;`

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

Note: After changing environment variables, if you had a console window 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 Applications

The procedure below builds both the Scribble and Asteroids Silverlight clients.

Note: You should build the service first, as described in the *Quick Start Guide* for the service, before following the procedure below.

If you have already built the C# or C++ sample service for Scribble, you do not need to follow the procedure below, as the Silverlight client will have already been built along with the service.

---

1. Navigate to the relevant directory:
    *   For Scribble:
        ```
        [Installed_directory]\SDK\Samples\Scribble\ScribbleClient
        ```
    *   For Asteroids:
        ```
        [Installed_directory]\SDK\Samples\Asteroids\AsteroidsClie
        nt
        ```
2. Double-click on the relevant file to open it in Microsoft Visual Studio:
    *   `ScribbleClient.sln`
    *   `AsteroidsClient.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.

You should now see a either a `ScribbleApp.xap` or an `AsteroidsApp.xap` file in the following folder:

```
[Installed_directory]\Server\webapp
```

3 Applications Overview

This chapter contains instructions on how to run the Scribble and Asteroids sample applications, and describes the functionality available in each.

Launching the Applications

The procedure for launching a Silverlight sample application is a two-step process which involves reloading the plugins in the server, then navigating to the client URL.

Reloading the Plugins

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

1. If the server is not running, launch it by double-clicking on the **Start PureWeb** desktop icon. This will automatically restart the plugins.

2. If the server is already running:
   a. Open a browser.
   b. Navigate to `http://localhost:8080/pureweb/config/plugins` (if applicable, replace `localhost:8080` with the correct server and port for your configuration of the PureWeb server).
   c. Enter `admin` into both the **Name** and **Password** fields and click the **Sign In** button.
   d. In the Configuration page, click the **Reload Plugins** button, under the **PureWeb Server Configuration** heading.

Navigating to the Client URL

The URL you use to launch the application depends on which service and client you want to run.

1. If the server is not running, launch it by double-clicking on the **Start PureWeb** desktop icon.

2. Open a browser window and navigate to the relevant client URL:

   C# Service:

   localhost:8080/app?name=ScribbleApp&client=silverlight&_diagnostics=true

C++ Service:

localhost:8080/app?name=ScribbleAppCpp&client=silverlight&_diagnostics=true

Java Service:

localhost:8080/pureweb/app?name=ScribbleAppJava&client=silverlight&_diagnostics=true

localhost:8080/pureweb/app?name=AsteroidsApp&client=silverlight&_diagnostics=true

---

Note: The `&_diagnostics=true` parameter is optional. See "Diagnostics Panel" on page 13 for more information.

Exercising the Scribble Functionality

The Scribble application is essentially a canvas that allows users to draw lines by left-clicking the mouse and dragging.
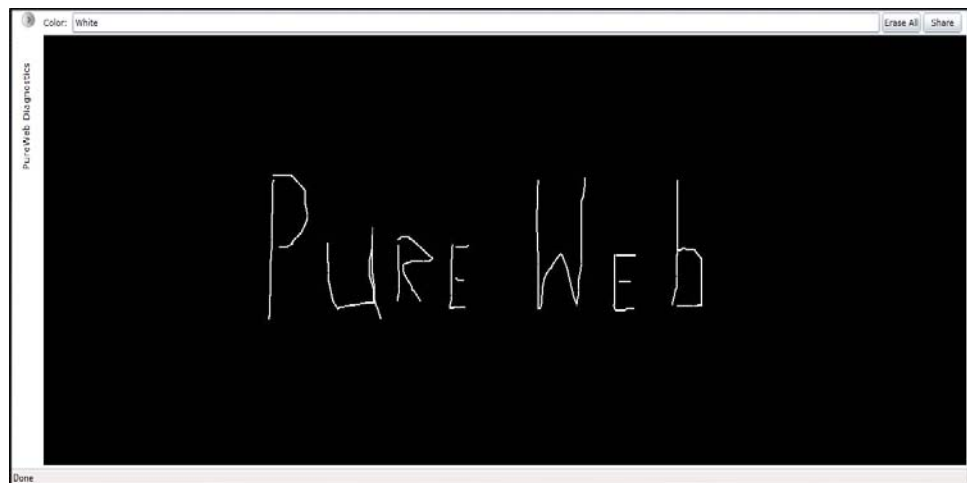
Scribble also offers the following functionality:

• 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.

• 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" on page 13).

• Share the application with other users by clicking the **Share** button (for more information, see section Sharing the Application, on page 14).

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

• When you resize the browser window, the service window resizes to the same size (works with the C# service only).

• 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.

- The **App State** tab displays an XML representation of the application's state.

- The **Bandwidth** tab is used to measure latency and bandwidth.

Exercising the Asteroids Functionality

The sample Silverlight Asteroids application is a PureWeb-enabled implementation of the classic video game with a twist -- it supports both single-player and two-player modes; each player independently controls his or her own ship.
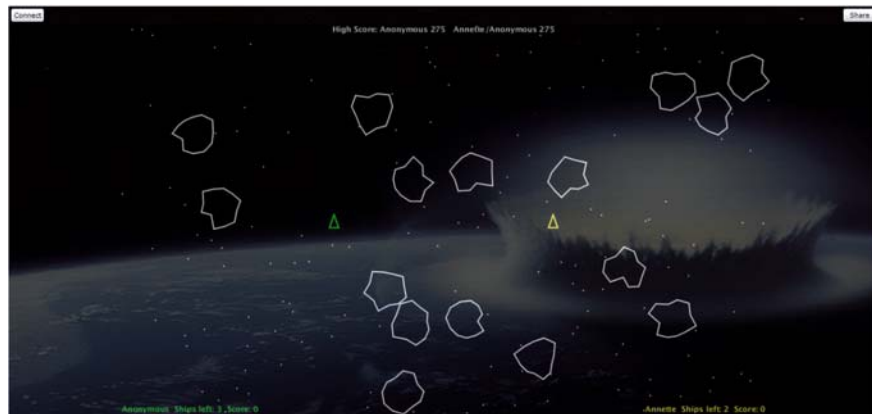
Single-Player Mode

To play Asteroids:

- Press any key to start the game.

- Use the up/down arrow keys to move forward/reverse.

- Use the left/right arrow keys to change your heading.

- Press the space bar to fire torpedoes.

- Activate shields by pressing the s key. Each player gets 5 seconds of shield time per game.

- Collide with an asteroid with your shields up to destroy the asteroid and collect the points.

Two-Player Mode

Before users can play in two-player mode, the first user must share the application with a second user, see Sharing the Application section, below.

Additional two-player mode functionality includes:

- Collide with an unshielded opponent with your shields up to destroy their ship and collect all of their points.

- Collide with a shielded opponent with your shields up to create an elastic collision.

Sharing the Application

Both the Scribble and Asteroids samples contain 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:

- Scribble 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 server's 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 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 list. All possible clients are listed, but only the clients that have actually been 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 Silverlight clients; 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 service-side code, please refer to the *Quick Start Guide* for the service programming language you want to work with.

You'll find the files described in this chapter at the following locations:

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

Scribble Client Code Pointers

The `MainPage.xaml.cs` file contains a number of methods and inner classes that are noteworthy for PureWeb enablement.

Application_Startup

This method, found in the `App.xaml.cs` file, provides the code that connects to the server. This file also contains event handlers and the code for error handling.

color_LostFocus

This method listens for focus lost events on the color text field. When this happens, the `ScribbleColor` application state property is set to the current color.

clear_Click

This method listens for buttons clicks on the **Erase All** button. When the button is clicked, a `Clear` command is sent to the service application. The service responds by clearing previously drawn scribbles, and sending an updated blank image back to the client.

share_Click

This method calls `Framework.Instance.Client.GetSessionShareUrlAsync`, which creates a collaborative URL that is displayed in a message box.

Asteroids Client Code Pointers

The `App.xaml.cs` and `MainPage.xaml.cs` files contain a number of methods and inner classes that are noteworthy from a PureWeb enablement perspective.

Application_Startup

This method, found in the `App.xaml.cs` file, provides the code that connects to the server. This file also contains event handlers and the code for error handling.

MainPage

This method, found in the `MainPage.xaml.cs` file, adds a handler that calls `OnLevelChanged` when the `/Level` changes in the application state.

connect_Click

This method is invoked when the **Connect** button is clicked. It adds a handler to call `ConnectSession` when the window is closed.

share_Click

This method is invoked when the **Share** button is clicked. The method calls the `ShowCollaborationDialog` method which will display in the Share URL dialog.

5 Debugging

Using the sample application as a model, this section describes how you would debug a PureWeb Silverlight client in Microsoft Visual Studio.

The information in this section is based on Microsoft Visual Studio 2010; it may vary slightly if you are using a different version.

For instructions on how to debug the service, please refer to the applicable PureWeb service *Quick Start Guide*.

Note: Before getting started:
- You should make Microsoft Internet Explorer your default browser when debugging Silverlight applications using Microsoft Visual Studio.
- If working in Microsoft Internet Explorer 8 or 9, deselect the **Enable Protected Mode** option found in the menu **Tools | Internet Option | Security**, then restart the browser.

1. Open the client solution file (`ScribbleClient.sln` or `AsteroidsClient.sln`) in Microsoft Visual Studio.

2. Select the project (`ScribbleClient` or `AsteroidsClient`).

3. From either the Project menu or the contextual (right-click) menu, select **Set as Startup Project**.

4. Select `TestPage.html` from the chosen project then, from either the Project menu or the contextual menu, select **Set as Start Page**.

   Note that `testpage.html` uses localhost, which means that if you are debugging collaboration, you will need to modify the html page to contain the IP address or DNS name of the machine.

5. If the PureWeb server is not currently running, start it by double-clicking the **Start PureWeb** desktop icon.

6. Go to the **Debug** menu and select the **Start Debugging** option, or press F5. Your default web browser (Microsoft Internet Explorer) will open with the test page displayed.

7. Launch the application using the relevant link, depending on which client and service you are working with:

- ◆ Launch Scribble Demo .NET with diagnostics
- ◆ Launch Scribble Demo C++ with diagnostics
- ◆ Launch Scribble Java with diagnostics
- ◆ Launch Asteroids Silverlight Client

8. If prompted for login information, type `admin` in both the **Name** and **Password** fields, then click the **Sign In** button.

9. Set a breakpoint in the relevant method of your sample client's `MainPage.xaml.cs` file:

- ◆ `OnScribbleColorChanged` (for Scribble)
- ◆ `AsteroidsView_MouseOver` (for Asteroids)
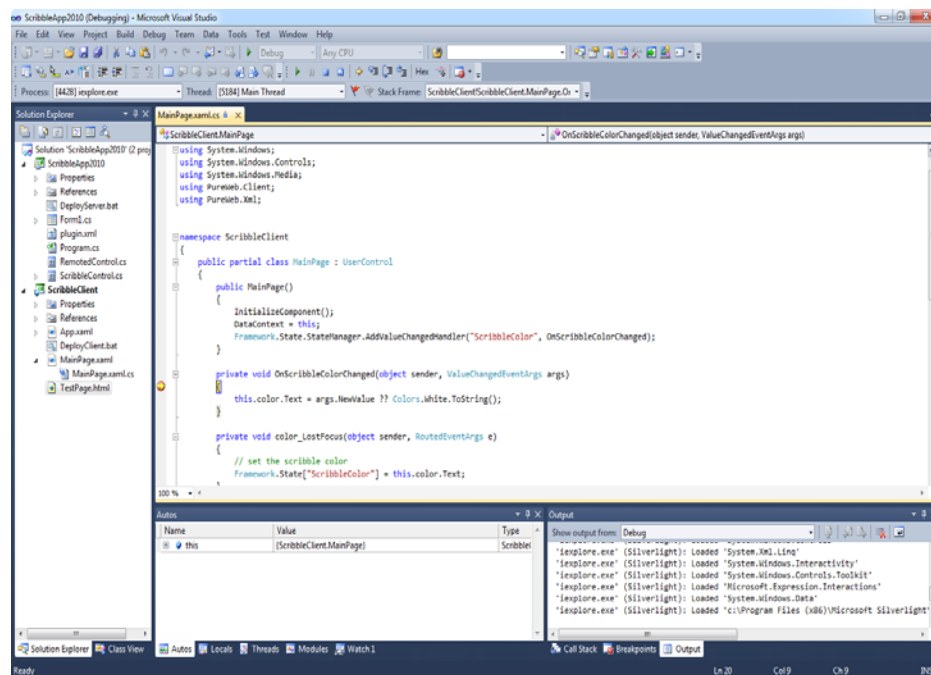
10. Trigger the breakpoint using the appropriate step below according to your sample client:

- ◆ For Scribble: enter a new color in the **Color** text box of the client application.
- ◆ For Asteroids: Move your mouse in the client application.