CALGARY SCIENTIFIC

PureWeb® STK 4.0

Quick Start Guide: Flex 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_FlexClient_QSG_07-2013_v1.000.00

Table of Contents

Preface

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

Intended Audience

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

• a working knowledge of Adobe Flash Builder
• an intermediate knowledge of Flex

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

The sample client is the Scribble application, a simple canvas that allows users to draw lines, change line color, and erase lines. It also allows collaboration.

Before working with the sample Flex client, you will need to build a sample service as described in the *Quick Start Guides* for the service development platforms.

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*
- Flex 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 to build a sample Flex 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 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 Flex sample client application, you will need:

- Adobe Flex 4.6 SDK
- Adobe Flash Player 11
- One of the following:
  - Adobe Flash Builder
  - Apache Ant

Setting the Environment Variables

The PureWeb server requires the `JAVA_HOME` environment variables; the Java home path must also be included in the `Path` variable.

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

1. Press the `Windows` key to bring up the **Start** menu (or **Start** screen in Windows 8).

2. Type "environment variables" to search that string.

3. Select the option **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.

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

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.

- 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 Java JDK:

1. In the list of system variables, select the `Path` variable.

2. Click on the **Edit...** button.

3. In the **Variable value** string, add the path to Java home if missing:
   `%JAVA_HOME%\bin;`

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

Note: Every entry in the `Path` system variable must be separated by a semi-colon.

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 PureWeb environment variables were set correctly (see "Setting the Environment Variables" on for instructions on how to access the Environment Variables dialog box):

    ◆ `PUREWEB_HOME`:
        `[Installed_directory]\Server`
    ◆ `PUREWEB_LIBS`:
        `[Installed_directory]\SDK\Redistributable\Libs`

Building the Sample Applications

You can build the sample Scribble application using either Adobe Flash Builder or Apache Ant.

Both procedures are described in this section; Flash Builder is usually the preferred method.

Note: Before completing the procedures below, you must build the service first, as described in the *Quick Start Guide* for each service development platforms.

Building with Adobe Flash Builder

1. Launch Adobe Flash Builder

2. Select **File | Import Flash Builder Project** from the main menu.

3. In the Import Flash Builder Project dialog box, choose the **Project Folder** option.

4. Click the **Browse** button and navigate to the following location:
    `[Installed_directory]\SDK\Samples\Scribble\ScribbleClientFlex`

5. Add the `PureWebSDK.swc` file to the project's libraries:

    a. In the Flash Builder's file navigation panel, expand the ScribbleClientFlex project file to display the `libs` folder.

    b. In Windows Explorer, navigate to the location of the `PureWebSDK.swc` file:

        `[Installed_directory]\SDK\Redistributable\Libs\Flex`

    c. Drag and drop the `PureWebSDK.swc` file from Windows Explorer into the `libs` folder for the project in Flash Builder.

6. Change the project's output folder:

    a. In the Flash Builder's file navigation panel, right-click the ScribbleClientFlex project file to display the contextual menu.

    b. Select **Properties**.

    c. In the Properties dialog box, select the **Flex Build Path** option.

    d. In the Flex Build Path dialog box, replace the value in the **Output folder** field with the following path; you can use the **Browse** button to navigate to the correct location:

        `[Installed_directory]\Server\webapp`

    e. Click **OK**. Flash Builder will close the Properties dialog box and build the project automatically.

Follow the instructions in next chapter, "Sample Applications Overview" on , for instructions on how to run the sample application after it has been built.

Building with Apache Ant

In addition to the JAVA_HOME environment variable used by the PureWeb server, building with Ant also requires that the FLEX_HOME and ANT_HOME environment variables be set correctly.

• FLEX_HOME must point to the location where you installed version 4.6 of the Flex SDK

• ANT_HOME must point to the location where you installed Ant 1.8.1

In addition, the `Path` variable must include the path to FLEX_HOME and ANT_HOME. See section "Setting the Environment Variables" on for instructions on how to set environment variables.

Build Procedure

1. Open a console (command prompt) window.

2. Navigate to the following directory:

    `[Installed_location]\SDK\Samples\Scribble\ScribbleClientFlex`

3. Type `ant` to run the `build.xml` build script.

Follow the instructions in next chapter, "Sample Applications Overview" on , for instructions on how to run the sample application after it has been built.

3 Sample Applications Overview

This chapter briefly describes the functionality available in the Scribble sample application.

Launching the Application

You can launch the Scribble Flex sample application from either the Adobe Flash Builder, or by navigating to a test page file. Both procedures are described in this section.

Reloading the Plugins in the Server

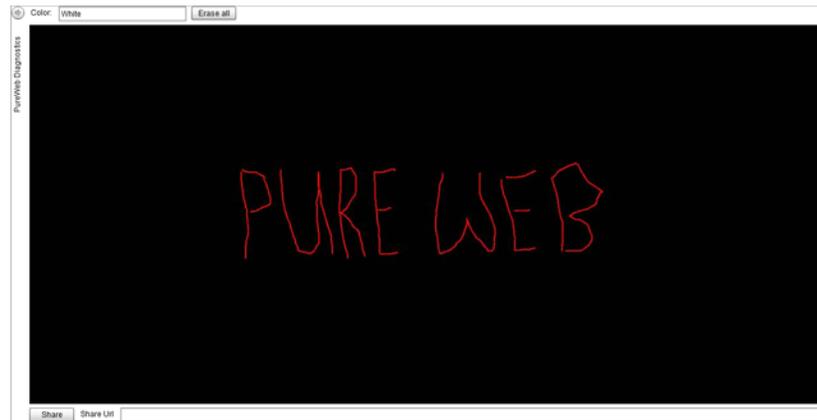This step is only necessary the first time you build the service application.

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.

Opening the Application

1. Navigate to the application link using either of the following three methods:
    *   In the PureWeb server: log in the server and in the menu links at the top, click the **Apps** link to go to the Applications page. In the Applications page, click the **Launch Flex** button associated with the Scribble application (the service application must already have been built).
    *   In Adobe Flash Builder: select the ScribbleClientFlex project in the Package Explorer and select **Run | Run** from the main menu.
    *   In Windows Explorer: navigate to the following location and double-click on the `TestPage.html` file:
        `[Installed_directory]\SDK\Samples\Scribble\ScribbleClientFlex`

2. Enter `admin` in both the name and password fields and click the **Sign In** button.

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 expand/contract button located in the upper-left corner of the client window (this is assuming that you set the diagnostics parameter to true in the application URL; for more information, see the "Diagnostics Panel" on),
*   share the application with other users by clicking the Share button (for more information, see section "Sharing the Application" on).

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.

To access the panel, add the following parameter to the client application's URL: `&_diagnostics=true`

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

- The **Options** tab is used to define options that impact image quality and bandwidth.
- The **Trace** tab is used to examine the messages to and from the server.
- The **App State** tab displays an XML representation of the application's state. Click **Refresh** to update the view with the latest state.
- The **Bandwidth** tab is used to measure latency and bandwidth.

Sharing the Application

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, 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 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 in the lower-left corner of the Scribble client application; this will display a URL in the **Share Url** field.

2. Copy the displayed Share URL 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; it's the best way to learn and to quickly become proficient with PureWeb.

This chapter gives some pointers about the code in the sample Flex client; 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 each service development platform.

ScribbleView Class

The `ScribbleClient.mxml` file contains the code that creates the `ScribbleView`, which is the main application class for this client. It includes:

- a Diagnostics view — displays the diagnostics information
- a Share button — allows you to share the session
- an Erase All button — erases all of your scribbles
- a Color input field — displays the value for ScribbleColor stored in application state

ScribbleClientViewModel Class

The `ScribbleClientViewModel.as` file contains the `ScribbleClientViewModel` class. It contains the functionality that allows the client to interact with the PureWeb service, including functions for getting and setting the framework instance.

connect

The `connect` function is called to connect you to the server.

updateColor and onScribbleColorChanged

The color used to draw lines on the scribble view is stored in the application state under the path `/ScribbleColor`.

The `updateColor` method sets the application state value, and `onScribbleColorChanged` is called by the PureWeb STK whenever the application state value changes.

clear

When you click on the **Erase All** button, a `Clear` command is sent to the service application. The service application responds by clearing previously drawn scribbles, and sends an updated blank image back to the client.

createShare and onShareRequest

When you click on the **Share** button, the client creates an `AppShare` service request and queues it for execution by the PureWeb server.

The `AppShare` service request creates a collaboration session for the application. The URL that is returned from the request can be shared with others to allow them to interact with the same application session. It also creates an event listener to monitor the share request.

5 Debugging

Using the sample application as models, this section describes how you would debug a PureWeb Flex client in Adobe Flash Builder.

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

Note: Google Chrome includes a plug-in of Flash Player that can not be used to debug Flex applications. For instructions on how to debug Flex application in Google Chrome see:
http://cookbooks.adobe.com/post_Debug_Flex_Applications_using_Google_Chrome-17271.html

---

The procedure below first describes how to configure the project for debugging in Adobe Flash Builder. It then describes how to set and trigger breakpoints.

Configuring the Scribble Project for Debugging

The procedure below will configure the run and the debug settings for the Scribble sample project in Adobe Flash Builder:

1. Right-click the **ScribbleClientFlex** project and select **Properties** from the menu.

2. Select **Run/Debug Settings** from the Project Properties menu.

3. Click **New** to open the Select Configuration Type dialog.

4. Select **Web Application** and click **OK** to edit the launch configuration properties.

5. Uncheck the **Use default** check box in the URL or path to launch section.

6. Browse to the sample application's test page:
    `[Installed_location]\SDK\Samples\Scribble\ScribbleClientFlex\TestPage.html`

7. Click **OK**.

8. Click **OK** again to close the Run/Debug Settings dialog.

Setting and Triggering Breakpoints

1. Open Adobe Flash Builder if it is not already open.

2. Set a breakpoint in the `eraseAllBn_click` method in the `ScribbleClientImpl.as` file.

3. Select the ScribbleClientFlex project in the Package Explorer and select **Run | Debug** from the main menu.

4. Click on the **Launch Scribble Demo.Net with diagnostics** link.

5. Enter `admin` in both the name and password fields.

6. Click the **Sign In** button to launch the application.

7. Draw lines on the canvas by left-clicking the mouse and dragging.

8. Click the **Erase All** button in the Scribble client to trigger the breakpoint.